DEC STD 065-0
Digital Standard Data Definitions

DOCUMENT IDENTIFIER: A-DS-EL00065-00-0 Rev D, 06-Mar-1991

ABSTRACT: This document defines the requirements for the definition of sharable data objects, including data elements, fields, data entities, relationships, and data models.

APPLICABILITY: These requirements apply to all individuals involved with the creation, management, or implementation of sharable data definitions.

STATUS: APPROVED 06-Mar-1991; use VTX SMC for current status.

This document is confidential and proprietary, and is the property of Digital Equipment Corporation. It is an unpublished work protected under the Federal copyright laws.

© Digital Equipment Corporation. 1985, 1991. All rights reserved.

DEC STD 065-0 Digital Standard Data Definitions

DOCUMENT IDENTIFIER: A-DS-EL00065-00-0 Rev D, 06-Mar-1991

REVISION HISTORY:	Rev A,	28-Feb-1985	
	Rev B,	06-Sep-1985	ECO Number APO01
	Rev C,	05-Sep-1986	ECO Number LJO02
	Rev D,	06-Mar-1991	ECO Number CTS03

Document Management Category:	DIS Data Management (TSD)
Responsible Department:	Data Resource Management (DRM) Program Office
Responsible Person:	Jean Reiczyk
Author:	David Crowley
SMC Writer:	Eric Falkof

APPROVAL: This document has been reviewed and recommended for approval by the General Review group for its category.

Jean Reiczyk, Enterprise Data Board

Eric Williams, Standards Process Manager

Direct requests for further information to:

David Crowley

MLO3-2/H14, DTN: 223-8504, MILPND::CROWLEY

Use VTX SMC to order copies of this document from Standards and Methods Control. Send distribution questions to JOKUR::SMC or call DTN: 234-4423.

Contents

CHAPTER 1 INTRODUCTION 1–1

1.1 PURPOSE OF THIS STANDARD 1–1

1.2 POLICY 1–2
- 1.2.1 APPLICABILITY 1–3
- 1.2.2 DEPENDENCIES AND RELATED POLICIES 1–3

1.3 SCOPE 1–4
- 1.3.1 TYPES OF MODELS COVERED BY THIS STANDARD 1–4
- 1.3.2 TYPES OF MODELS NOT COVERED BY THIS STANDARD 1–4

1.4 RESPONSIBILITIES 1–5
- 1.4.1 ENTERPRISE DATA BOARD 1–5
- 1.4.2 CONTROLLERS 1–5
- 1.4.3 INFORMATION SYSTEMS MANAGERS 1–6
- 1.4.4 DATA RESOURCE SPECIALISTS 1–6

1.5 SUMMARY OF REQUIREMENTS 1–7
- 1.5.1 DATA MODELS 1–7
- 1.5.2 ENTITIES 1–9
- 1.5.3 ASSOCIATIONS 1–11
- 1.5.4 ELEMENTS 1–13
- 1.5.5 FIELDS 1–15

1.6 BACKGROUND INFORMATION 1–17
- 1.6.1 CHANGES FROM THE PREVIOUS REVISION 1–17
- 1.6.2 DEFINITION OF TERMS USED IN THIS DOCUMENT 1–17
- 1.6.3 DOCUMENT STRUCTURE 1–19

CHAPTER 2 STANDARD DATA MODELS 2–1

2.1 INTRODUCTION TO THE DATA MODELS 2–1

2.2 SYMBOLS ON THE MODELS 2–1

2.3 CARDINALITY 2–2

2.4 SUBTYPES 2–3

2.5 DEFINITIONS OF STANDARD DATA MODELS 2–5

STANDARD DATA MODEL REQUIREMENTS 2–6

TYPE HIERARCHY OF DATA OBJECTS 2–12

CHAPTER 3 STANDARD ENTITY DEFINITIONS 3–1

3NF ENTITY 3–2

3NF MODEL 3–4

ASSOCIATION 3–6

BUSINESS DATA OBJECT 3–8

BUSINESS TERM 3–10

CLASSIFICATION 3–12

CODE SET 3–14

CODE VALUE 3–16

COMPLEX RELATIONSHIP 3–18

CONTROLLER 3–21

DATA DESCRIPTION LANGUAGE 3–23

DATA FORMAT 3–25

DATA MODEL 3–27

DATA OBJECT 3–29

DATE FORMAT 3–31

ELEMENT 3–32

ENTITY 3–34

FIELD 3–36

IMPLEMENTATION DATA OBJECT 3–38

NONSTANDARD FIELD 3–40

NUMERIC FORMAT 3–42

OPTIMIZED ENTITY 3–44

OPTIMIZED MODEL 3–46

RELATED NAME 3–49

RELATED NAME CONTEXT 3–51

RELATIONSHIP 3–52

SECURITY CLASSIFICATION 3–54

SIMPLE RELATIONSHIP 3–55

STANDARD FIELD 3–57
SUBTYPING 3–59
TEXT FORMAT 3–61

CHAPTER 4 STANDARD ASSOCIATION DEFINITIONS 4–1
AGGREGATE OF 4–2
ASSOCIATES 4–4
BASED ON 4–6
CLASSIFIED BY 4–8
CONSISTS OF (CODE SET) 4–10
CONSISTS OF (DATA MODEL) 4–12
CONTROLLED BY 4–14
CREATED FOR 4–16
DEFINED AS 4–18
HAS ATTRIBUTE (COMPLEX RELATIONSHIP) 4–21
HAS ATTRIBUTE (ENTITY) 4–23
HAS DATA FORM 4–25
HAS DATA SECURITY 4–27
HAS METADATA SECURITY 4–29
IS AN ASSOCIATION SUBTYPE 4–31
IS A BUSINESS DATA OBJECT SUBTYPE 4–33
IS A DATA FORMAT SUBTYPE 4–35
IS A DATA MODEL SUBTYPE 4–37
IS A DATA OBJECT SUBTYPE 4–39
IS AN ENTITY SUBTYPE 4–41
IS A FIELD SUBTYPE 4–43
IS AN IMPLEMENTATION DATA OBJECT SUBTYPE 4–45
IS A RELATIONSHIP SUBTYPE 4–47
KNOWN AS 4–49
NAMED USING 4–51
PARTICIPATES IN PRIMARY KEY OF 4–53
REPLACES 4–55
SUBCLASS OF 4–57
SUBSET OF 4–60
TRANSFORMS INTO 4–62
VALUES BOUNDED BY 4–64

CHAPTER 5 STANDARD ELEMENT DEFINITIONS 5–1
ABBREVIATED NAME 5–2
AGGREGATION NARRATIVE 5–4
BUSINESS NAME 5–6
BUSINESS TERM 5–8
BUSINESS TERM ABBREVIATION 5–10
BUSINESS TERM DEFINITION 5–12
CODE SET NAME 5–14
CODE DESCRIPTION 5–16

CODE VALUE 5–18
COMPUTED-BY EXPRESSION 5–20
COMPUTED-BY ROUTINE NAME 5–22
CONTROLLER NAME 5–24
CONTROLLER CONTACT EMPLOYEE IDENTIFIER 5–26
DATA DESCRIPTION LANGUAGE NAME 5–28
DATA FORMAT TYPE 5–30
DATA OBJECT IDENTIFIER 5–32
DATA OBJECT REVISION 5–34
DATA OBJECT STATUS 5–36
DATA OBJECT TYPE 5–38
DEFAULT ACTION NARRATIVE 5–40
DERIVATION NARRATIVE 5–42
DESCRIPTION 5–44
DOCUMENTATION SOURCE CITATION 5–46
ELEMENT EDIT STRING 5–48
ELEMENT SCALE 5–50
FIELD DATATYPE 5–52
FIELD EDIT STRING 5–54
FIELD HELP 5–56
FIELD JUSTIFICATION 5–58
FIELD LABEL 5–60
FIELD NAME 5–62
FIELD QUERY NAME 5–64
FIELD SCALE 5–66
FIELD SIZE 5–68
INITIAL VALUE 5–70
INPUT VALIDATION EXPRESSION 5–72
INPUT VALIDATION ROUTINE NAME 5–74
INTEGRITY NARRATIVE 5–76
MAXIMUM CARDINALITY 5–78
MAXIMUM CHARACTER COUNT 5–80
MINIMUM CARDINALITY 5–82
MISSING VALUE 5–84
NOTES 5–86
RELATED NAME 5–88
RELATED NAME CONTEXT 5–90
RELATIVE DATE INDICATOR 5–92
REPLACEMENT NARRATIVE 5–94
REVERSE RELATIONSHIP NAME 5–96
SECURITY CLASSIFICATION NAME 5–98
SIGNED NUMERIC INDICATOR 5–100
TEXT JUSTIFICATION 5–102
TRANSFORMATION NARRATIVE 5–104
VALUE TYPE INDICATOR 5–106
VARIABLE SIZE INDICATOR 5–108

APPENDIX A REFERENCED DOCUMENTS A–1

A.1 EL-CLASS DIGITAL DOCUMENTS A–1

A.2 OTHER DIGITAL DOCUMENTS A–1

INDEX

FIGURES

2–1 Sample Modeling Symbols for Relationships 2–2
2–2 Sample Modeling Symbols for Subtyping 2–3
2–3 Standard Data Model Requirements 2–8
2–4 Type Hierarchy of Data Objects 2–14

TABLES

4–1 Correspondence of Standard Field to Element 4–19
5–1 Data Format Type Table 5–30
5–2 Data Object Status Table 5–36
5–3 Data Object Type Table 5–38
5–4 Relative Date Indicator Table 5–92
5–5 Security Classification Name Table 5–98
5–6 Signed Numeric Indicator Table 5–100
5–7 Text Justification Table 5–102
5–8 Value Type Indicator Table 5–106
5–9 Variable Size Indicator Table 5–108

1 INTRODUCTION

Digital's internal information systems establish the infrastructure that permits sharing of information and expertise from the point of creation to the point of use, within the corporation into the extended enterprise.

Data management is the discipline that Digital uses to integrate this infrastructure: the businesses with systems, and the systems with each other. Data management provides techniques to describe and communicate information about business functions easily and consistently. Data management also provides the techniques for translating this information about the business into systems designs.

Digital has chosen data management as the focal point for the integration of systems and businesses because it ensures that all parties in the corporation are dealing with the same facts. The data can be analyzed, manipulated, or redistributed in a variety of ways, always responding to the latest business and organizational strategies. But throughout any of these changes, the data must continue to mean the same thing and represent the same facts.

One of the principal techniques in data management is data modeling. Data models are graphic representations of the information used by a business, which are complemented by definitions of all the objects that appear on the diagram.

Data modeling is a principal feature of modern systems design methodologies. Data modeling is the technique that allows systems designers first to understand the information requirements of the business, and then to design data bases that meet these requirements. The reason for this standard is to establish the requirements for defining data objects so that they can be shared and reused in many systems design projects; its purpose is not to define the methods of analysis and design, but rather to make the results sharable.

1.1 PURPOSE OF THIS STANDARD

This standard defines the minimum requirements for the definitions of data models and related data objects. Data definitions that comply with these requirements become *standard data definitions*. These standard data definitions must then be used throughout the corporation in the design and implementation of information systems. The objectives of standardizing data definitions are the following.

- Provide sharable, consistent models that can be used in multiple businesses, across functional or organizational boundaries.

- Provide the businesses with better control over the design of their information systems.
- Ensure that information systems are responsive to business change.
- Ensure consistent data regardless of the system or organization where that data appears.
- Facilitate the integration of systems, reflecting the integration of businesses.
- Promote efficient and flexible database design.
- Improve the productivity of systems developers.
- Enable the implementation of information delivery architectures.

This document establishes the framework in which models can be standardized regardless of the technique used to create them. There are many techniques and methodologies defined in the industry for analyzing a business and documenting a data model. To achieve consistency, this standard uses neutral terms whenever there is a conflict of definition among methodologies. This standard presently defines two forms of data model that may be standardized: models in Third Normal Form (3NF), and models that have been optimized for a particular application or implementation. Different methodologies use different names for these two forms of data model, but most methodologies include a step or deliverable that corresponds to each of these forms.

There are also many implementation languages that are used for defining data items. Again, this standard provides a mechanism for linking these different declarations. What these definitions have in common is that they implement a common piece of business information. This standard provides a method for defining that common piece of business information (called an Element), and to then define as many implementation structures (called Fields) as are required in whatever different data description languages are appropriate and necessary.

Through consistency of 3NF models, this standard establishes a basic foundation that supports the diversity of businesses across the corporation. Different businesses need to manipulate information in different ways, but this does not change the underlying structure of the information itself. Although authority is diffused throughout the corporation, the decisions about data can be made once at the designated place, and then shared among organizations.

1.2 POLICY

A Standard Data Definition is any definition of data that conforms to all the requirements of this standard, and which has been approved by a valid controller. Such definitions will be marked with a Data Object Status of *Approved*.

All data that supports Digital's operations must be defined and standardized in the context of a data model. All information systems development must include the standardization of an Optimized data model that is based on one or more integrated, standardized, 3NF data models. Refer to Digital Information Management and Technology (DIM&T) Policy 5.12 for definition of specific responsibilities and schedule requirements.

All approved System Life Cycle (SLC) methodologies must deliver data models that comply with the requirements of this standard. Refer to DIM&T Policy 4.01 for definition of specific responsibilities.

1.2.1 APPLICABILITY

This standard applies to all data modeling activity within Digital. This standard must be used for the creation, approval, and implementation of all data models, and supporting data object definitions, within Digital.

This standard presumes knowledge of data modeling and data analysis. Readers of this document should be familiar with at least one data modeling technique. This standard does not include any detailed explanation of how to analyze the data in a business, nor how to effectively model that data.

1.2.2 DEPENDENCIES AND RELATED POLICIES

Refer to Appendix A for complete information about any related documents, including the full title and order numbers where applicable.

The *Data Description and Naming Guidelines* document, referred to throughout this standard as the *DDNG*, describes the requirements and procedures that must be used in the formation of Business Names, Descriptions, Field Names, and Abbreviated Names of data objects. DEC STD 065-0 requires that the DDNG be used in the development of all standard data object definitions.

Digital Information Management and Technology (DIM&T) Policies 4.01 and 5.12 defines the responsibilities and requirements within the IM&T function for the implementation of this standard, and for conformance with approved, standardized, data definitions.

The procedures that must be used to certify that a data object definition complies with this standard are the responsibility of the Enterprise Data Board (EDB). The authority to conduct such certification review is delegated to individual operating groups, along with certain mandatory procedures for development, review, and quality assurance of the definitions. These procedures are available from the EDB.

Note: The procedures for review and approval of data object definitions are still under development as of the publication of this standard. When these procedures are published, a copy will be distributed to all holders of this standard.

1.3 SCOPE

This standard establishes the requirements for the definitions of *data models, entities, associations, elements*, and *fields*. These are the principal objects used in the definition and implementation of business data. There are other data object types that can be defined, especially in the context of an implementation (for example, records), but that are not addressed in this revision of the DEC STD 065-0. These other data object types will be the subject of future extensions of this standard.

This standard defines the minimum requirements for the definitions of data models and related data objects. Specific Systems Life Cycle methodologies may establish requirements beyond this minimum.

1.3.1 TYPES OF MODELS COVERED BY THIS STANDARD

This standard addresses two categories of data models: models in third normal form (3NF models), and the optimized versions of these 3NF models (optimized models). These names refer to the stage of analysis and design in which these categories of data model are produced.

Different methodologies use different terms to refer to the data models developed within the methodology. For example, the Merise technique for data analysis (as described in *DMR Life Cycle System Development Guides*) uses the term Conceptual Data Model to refer to a model in third normal form, and the term Functional Data Model to refer to the optimized version of that 3NF model. Other modeling techniques use their own terms as well; but these terms do not have a universal meaning. For example, in one methodology the term "Conceptual Model" refers to a high-level, pre-normalized business model, and the term "Logical Model" refers to the third normal form version of that model. But in another methodology, the term "Conceptual Model" refers to the 3NF level, and yet another term refers to a prenormalized version. Therefore, this standard adopts the terms *3NF* and *optimized* to describe the models themselves, rather than to position them within a prescribed methodology.

Note that each analytic methodology embeds many other conditions and concepts into its definition of modeling terms. However, what remains constant across most data analysis methodologies is that they include both a model in third normal form, and a version of that model that has been optimized for access, performance, ease-of-use, and/or security.

1.3.2 TYPES OF MODELS NOT COVERED BY THIS STANDARD

The techniques of data analysis and business analysis include many other types of models that are not covered by this standard. The following list provides a sample of the types of models not covered by this standard.

- Conceptual data models that are not yet normalized into 3NF
- Subject area data models that show general links between subject areas

- Physical data models that show the implementation of data structures as files, records, indexes, and views
- Data flow diagrams or other process models that define processes and the data that is utilized by a process

1.4 RESPONSIBILITIES

This section identifies the organizations and functions that have specific responsibilities established by this standard.

1.4.1 ENTERPRISE DATA BOARD

The EDB is the governing body for DEC STD 065-0. The EDB is responsible for ensuring that DEC STD 065-0 is maintained. All changes, extensions, or future revisions of DEC STD 065-0 must be approved by the EDB.

The EDB is responsible for establishing the procedures to be used to produce, share, approve, and maintain data models and data object definitions. The EDB is responsible for ensuring that the tools used to collect, manage, and distribute data models and data definitions support DEC STD 065-0.

The EDB is also responsible for ensuring that all related standards and guidelines are consistent with this standard.

The EDB is responsible for the proper application and enforcement of this standard. The EDB oversees the methods used in local organizations relative to creating and integrating data models. The EDB is responsible for identifying areas of noncompliance and for correcting these problems.

1.4.2 CONTROLLERS

The controller is the agent or agency within the corporation that has the knowledge, authority, and responsibility to ensure the accuracy and usability of the definition of a business data object. For each data object, there is only one controller. The controller can be identified as a specific job title, a position, a particular responsibility within an appropriate organization, or a chartered cross-functional or cross-organizational board or committee. Controllership is a dimension of overall business responsibility for a particular aspect of the corporation's operations and strategies.

The controller is responsible for ensuring that the definition of a data object is accurate and supports the requirements of the business.

The controller of a definition may assign an individual employee to be the contact point to execute any of the specific responsibilities of controllership for that definition, and for the resolution of issues and questions about the definition.

The controller of a definition may review, and may disapprove, definitions of any other business data objects that are derived from their controlled data object, in order to ensure that they are consistent with the intent of the original. For example, the controller of the enterprise-wide definition of Trading Partner may review any model that uses Trading Partner and may disapprove if the proposed model violates the intent of the original definition. This includes specifically: models that contain the controlled entity or relationship; entities that are subtypes or classifications of a controlled entity; elements that are subclasses of a controlled element; and any object that is a transformation, aggregation, or replacement of a controlled object.

The controller is responsible for ensuring that the definition is maintained. The controller is responsible for reviewing changes to the definitions of related data objects and ensuring that their controlled definition responds to changes in business requirements.

The controller of an element is responsible for the field definitions for that element, including any nonstandard field definitions. The controller may delegate the authority for review of field definitions to an appropriate technical specialist.

The controller is not responsible for ensuring that the modeling technique has been used properly. This responsibility is fixed by the System Life Cycle being utilized, and by the policies of the implementation of that life cycle within the organization. The controller is not responsible for ensuring that the definition adheres to the requirements of this standard. This is a responsibility of the EDB, as defined in Section 1.4.1, and the individual Data Resource Specialist, as defined in Section 1.4.4.

1.4.3 INFORMATION SYSTEMS MANAGERS

Managers of information systems organizations must ensure that all system development uses an approved System Life Cycle, and that the data models and definitions produced in their organization comply with this standard and with the procedures for approving standardized data definitions. They must ensure that models and systems developed in their organization use existing standardized data object definitions where appropriate.

1.4.4 DATA RESOURCE SPECIALISTS

The actual work of modeling, creating and maintaining data object definitions, and implementing data structures is done by people with a broad variety of job titles and responsibilities. Some of the typical job titles include Business Systems Analyst, Data Analyst, Systems Analyst, Data Base Administrator, Data Administrator, Dictionary Administrator, Information Systems Specialist, Programmer, or Software Engineer.

For the purpose of this standard, a Data Resource Specialist is defined as any individual who is responsible for creating, managing, or implementing any data object definition. These responsibilities can be assigned to a member of the development project team; to an organization's Data Base Administrator; to a dedicated data management group; or they can be split among any of the above. A Data Resource Specialist is not a new role, nor a specific job class; it is a title applied to anyone who performs some portion of the work described above.

Data Resource Specialists are responsible for ensuring that any definitions they create comply with this standard, and with the procedures for standardization as defined by the EDB. These individuals are responsible for maintaining the ongoing consistency of their definitions with other applicable definitions. They are responsible for using existing standardized data object definitions where appropriate. And Data Resource Specialists are responsible for implementing data structures that are consistent with approved models and definitions.

1.5 SUMMARY OF REQUIREMENTS

This section provides a summary of the rules and requirements that are defined in this document. In all cases, additional information about the requirements for any piece of a definition can be found in the other chapters of this document.

Standardization procedures or local requirements may establish additional requirements, beyond those listed here. For example, the procedure may require that the name of the person who created the data model be recorded, or the date on which a definition was approved.

1.5.1 DATA MODELS

This section summarizes the rules and requirements that apply to the definition of standardized data models. Where there are differences between the requirements of 3NF Models versus Optimized Models, these differences are mentioned within the item where the difference is.

1 Every Data Model must be defined per the requirements for any type of Business Data Object, as follows:

a. Every Data Model must have a Description, a Business Name, and an Abbreviated Name that conform to the requirements of the DDNG. All the Business Terms in the Business Name must be defined in the Glossary of Business Terms, and the standard abbreviations of those terms must be used in creating the Abbreviated Name, following the rules defined in the DDNG.

b. The Data Object Type must be either *3NF Model* or *Optimized Model*.

c. The Integrity Narrative must include a description of the scope of the model. For Optimized Models, the Integrity Narrative must state the environmental constraints, implementation technology, performance requirements, and security constraints that establish the basis for the optimization.

d. The Notes may be used to provide additional information not included in the Description or Integrity Narrative.

e. A Controller must be identified. For 3NF Models, the controller is the agent who is chartered to define the business area within the scope of the model. For Optimized Models, the controller is usually the agent who has commissioned the project that will implement the model. An individual employee may also be designated as the contact point on behalf of the controller.

f. The Security Classification of both the data described by the model, and the model itself, must be identified. The classification must be at least as high as the classification of any piece of data (or metadata) contained in the scope of the model.

g. The model may be classified using defined Business Terms. Related Names may be added to assist future users in finding this model within a repository.

2 Every Data Model must include an Entity-Relationship Diagram that includes at least every Entity and every Association on the model. (Additional information, such as the names of keys or attributes, may be shown depending on the requirements of the methodology and tools employed to develop the model.)

3 Every Data Model must list all the Entities that are included in it. All Entities must themselves have standardized data definitions. The list of Entities may include a listing of each Entity's primary key and attributes, as they are defined in the Entity's standard definition. On a 3NF Model, every Entity must be a 3NF Entity.

4 Every Data Model must list all the Relationships that are included in it. All the Relationships must themselves have standardized definitions. The list of Relationships may include a listing of the attributes of any relationship, as they are defined in the Relationship's standard definition. On an Optimized Model, every Relationship must be a Simple Relationship.

5 The Relationships between this Data Model and other standardized Data Models must be documented, as follows:

a. If the model has resulted from a transformation of an existing model into a new business context or new level of abstraction then this must be documented with a Transformation Narrative. Every Optimized Model must identify the 3NF Model(s) upon which the optimization has been performed. The specific choices that were made in this optimization must be recorded as a Transformation Narrative.

b. If the model is a component of another model (for example, it is a Facet) then it must be documented with an Aggregation Narrative.

c. If the model replaces any other model, this replacement must be identified. The Data Object Status of the replaced model must become *Obsolete*. If the Data Model is a revision of an existing model, its Data Object Revision must be incremented by one above the previous revision. A Replacement Narrative may be supplied to describe the justification of the replacement or to provide guidance for implementing the upgrade.

6 Models that are not revisions of an existing model must be assigned a Data Object Identifier following current procedures of data administration. The Data Object Revision of new objects begins at 1. If the model is a revision, the Data Object Identifier is copied from the current version and the Data Object Revision is incremented by 1. All newly created data objects must have a Data Object Status of *Work in Process*.

1.5.2 ENTITIES

This section summarizes the rules and requirements that apply to the standardized definitions of entities. Where there are differences between the requirements of 3NF Entities versus Optimized Entities, these differences are mentioned within the item where the difference is.

1 Every Entity must be defined per the requirements for any form of Business Data Object, as follows:

a. Every Entity must have a Description, a Business Name, and an Abbreviated Name that conform to the requirements of the DDNG. All the Business Terms in the Business Name must be defined in the Business Terms Glossary, and the standard abbreviations of those terms must be used in creating the Abbreviated Name, following the rules defined in the DDNG.

b. The Data Object Type must be either *3NF Entity* or *Optimized Entity*.

c. The Integrity Narrative must define the rules that control the creation, modification, or deletion of this Entity. If the identification of the Entity depends on a relationship to another entity (that is, it is a weak entity), this dependence must be described.

d. The Notes may be used to provide additional information not included in the Description or Integrity Narrative.

e. A Controller must be identified. The authority of the controller must extend through all business domains where the Entity is used. An individual employee may also be designated as the contact point on behalf of the controller.

f. The Security Classification of both the data contained in the Entity, and the Entity definition itself, must be identified. The classification of the Entity must be at least as high as the classification of any of the Elements that are attributes of the Entity.

g. The Entity may be classified using defined Business Terms. Related Names may be added to assist future users in finding this Entity within a repository.

2 Every Entity must be included on at least one Data Model. If the Entity is not included on an approved data model, its Data Object Status cannot be *Approved*.

3 Every Entity definition must list all Elements that comprise its primary key. All such Elements must themselves have standardized element definitions. Across all approved definitions, no two 3NF Entities can have the exact same set of Elements as their primary key.

4 Every Entity definition must list all the Elements that are its attributes. All such Elements must themselves have standardized element definitions. Across all approved definitions, no Element can appear as an attribute of more than one 3NF Entity. The attributes of a 3NF Entity must meet the requirements for third normal form; this generally will exclude Elements that are derived from other Elements on the same model, and Elements that are a subclass of other Elements on the model. An Entity may have zero attributes.

5 The relationships between this Entity and other standardized data objects must be documented, as follows:

a. If the Entity is a classification or a subtype of another Entity, then it must have the same primary key as its parent Entity and its attribute list must include all the attributes of the parent Entity.

b. If an Entity results from the transformation of another Data Object, this must be described in a Transformation Narrative. An Optimized Entity that is a transformation of a Complex Relationship must include this linkage in its definition.

c. If the Entity is an aggregation of other Entities, then it must be documented with an Aggregation Narrative.

d. If this Entity replaces any other Data Object, this replacement must be identified. The Data Object Status of the replaced object must become *Obsolete*. If the Entity definition is a revision of an existing definition, its Data Object Revision must be incremented by one above the previous revision. A Replacement Narrative may be supplied to describe the justification of the replacement or to provide guidance for implementing the upgrade.

6 Entities that are not revisions of an existing Entity must be assigned a Data Object Identifier following current procedures of data administration. The Data Object Revision of new objects begins at 1. If the Entity is a revision, the Data Object Identifier is copied from the current version and the Data Object Revision is incremented by 1. All newly created Data Objects must have a Data Object Status of *Work in Process*.

1.5.3 ASSOCIATIONS

This section summarizes the rules and requirements that apply to the standardized definitions of Associations. Where there are differences between the requirements of Complex Relationships, Simple Relationships, Classifications, or Subtyping, these differences are mentioned within the item where the difference is.

1 Every Association must be defined per the requirements for any form of Business Data Object, as follows:

a. Every Association must have a Description, a Business Name, and an Abbreviated Name that conform to the requirements of the DDNG. All the Business Terms in the Business Name must be defined in the Glossary of Business Terms, and the standard abbreviations of those terms must be used in creating the Abbreviated Name, following the rules defined in the DDNG. In addition, a Reverse Relationship Name may be defined for relationships.

b. The Data Object Type must be either *Classification, Subtyping, Complex Relationship*, or *Simple Relationship*.

c. The Integrity Narrative must define the rules governing the creation and destruction of the Association. The Integrity Narrative must define any interdependency between this Association and other objects, such as an exclusivity with another Association or an integrity constraint that is governed by the value of some attribute. For Subtyping and Classification, the Integrity Narrative must identify the criteria by which occurrences of the supertype or superclass are to be assigned to the subtypes or classes.

d. The Notes may be used to provide additional information not included in the Description or Integrity Narrative.

e. A Controller must be identified. The controller must be the agent responsible for establishing the business rules that are embodied in the association. For Associations that are created on an Optimized Model, the controller will generally be either the controller of the root association from which the new Association was transformed, or the controller of the Optimized Model. An individual employee may also be designated as the contact point on behalf of the controller.

f. The Security Classification of both the occurrences of the Association and the definition of the Association must be identified. The classification of the occurrences must be at least as high as the classification of the entities associated.

g. The Association may be classified using defined Business Terms. Related Names may be added to assist future users in finding this Association within a repository.

2 Every Association must identify the Entities that are associated through it. For each Entity, the Minimum Cardinality and the Maximum Cardinality must be defined. These Entities must appear on all models that contain the Association.

3 If a Complex Relationship has attributes, these Elements must be identified. The Elements themselves must have standardized Data Object definitions. These Elements cannot appear as attributes of any other entity on this model, or on any other Complex Relationship anywhere. If these Elements appear on another 3NF Entity from some other model, then the Primary Key of that 3NF Entity must be equivalent to the concatenation of the Primary Keys of the Entities that are associated by the Complex Relationship on which the attribute appears.

4 The relationships between this association definition and other standardized Data Objects must be documented.

a. Any Complex Relationship that is transformed into a combination of Simple Relationships and/or Entities during optimization must be documented with a Transformation Narrative.

b. If this Association replaces any other Data Object, this replacement must be identified. The Data Object Status of the replaced object must become *Obsolete*. If the Association definition is a revision of an existing definition, its Data Object Revision must be incremented by one above the previous revision. A Replacement Narrative may be supplied to describe the justification of the replacement or to provide guidance for implementing the upgrade.

5 Associations that are not revisions of an existing Association must be assigned a Data Object Identifier following current procedures of data administration. The Data Object Revision of new objects begins at 1. If the Association is a revision, the Data Object Identifier is copied from the current version and the Data Object Revision is incremented by 1. All newly created Data Objects must have a Data Object Status of *Work in Process*.

1.5.4 ELEMENTS

This section summarizes the rules and requirements that apply to the standardized definitions of Elements. These requirements do not change, regardless of whether the Element is used in a 3NF context or in an Optimized context.

1 Every Element must be defined per the requirements for any form of Business Data Object, and must also meet additional definitional requirements, as follows:

a. Every Element must have a Description, a Business Name, and an Abbreviated Name that conform to the requirements of the DDNG. All the Business Terms in the Business Name must be defined in the Business Terms Glossary, and the standard abbreviations of those terms must be used in creating the Abbreviated Name, following the rules defined in the DDNG. The Abbreviated Name will be used as a standard Field Name.

b. The Data Object Type must be *Element*.

c. The Integrity Narrative must describe the rules for creation or modification of values for this Element. This includes any requirements for form or content beyond what is expressed in the Data Format or Code Set for this Element (see below). The Integrity Narrative will be used as the specification for the Input Validation procedures for fields that contain this Element.

d. The Notes may be used to provide additional information not included in the Description, Integrity Narrative, Derivation Narrative, or Default Action Narrative.

e. A Controller must be identified. The authority of the Controller must extend through all business domains where the Element is used. An individual employee may also be designated as the contact point on behalf of the Controller.

f. The Security Classification of both the values of the Element and the Element definition itself must be identified.

g. If the Element has some interdependence with the values of other elements, this must be described in a Derivation Narrative. This narrative will be used as the specification for the Computed-by procedures for Fields that contain this Element, if the interdependence is implemented as a procedure.

h. Any rules regarding how to handle the absence of a value for the Element must be described in a Default Action Narrative. This narrative will be used as the specification for the Missing Value and Initial Value of Standard Fields for this Element.

i. The Element may be classified using defined Business Terms. Related Names may be added to assist future users in finding this Element within a repository.

2 If an Element refers directly to a subset of the possible values of an existing Element, it must be defined as a subclass of that parent Element and it must not have its own Data Format. Otherwise, every Element must have a Data Format with the following information defined:

   a. Elements that are used in computation must have a Data Format Type of *Numeric*. This data format must include a definition of the Element Edit String, that will be used as the standard Field Edit String; of the Maximum Character Count, that will be used as the standard Field Size; of the Element Scale, that will be used as the standard Field Scale. The Signed Numeric Indicator must indicate whether or not the Element can have negative numbers as values.

   b. Elements that represent an absolute point in time or a relative period of time must have a Data Format Type of *Date*. This Data Format must include a definition of the Element Edit String, and must indicate whether the data represents an absolute date/time or a relative date/time. The Maximum Character Count of a *Date* should be left blank.

   c. Elements that contain noncomputational data, nondate data, or which have a variable length must be defined with a Data Format Type of *Text*. This Data Format must include a definition of the Element Edit String, that will be used as the standard Field Edit String; of the Maximum Character Count, that will be used as the standard Field Size; and of the Text Justification, that will be used as the standard Field Justification. Variable-size elements must be indicated as such.

3 Every Element must appear in at least one role (that is, as an attribute or as part of a key) in at least one Entity or Relationship on at least one Data Model. No Element can be approved outside the context of an approved Entity or Relationship, which in turn appears on an approved Data Model.

4 The relationships between this Element and other standardized Data Objects must be documented, as follows:

   a. If this Element is composed of some combination of other Elements (sometimes referred to as a Group or Aggregate Element), then any ordering rules, special formatting rules, or value constraints must be defined in an Aggregation Narrative.

   b. If this Element replaces any other Element, this replacement must be identified. The Data Object Status of the replaced Element must become *Obsolete*. If the Element is a revision of an existing Element, its Data Object Revision must be incremented by one above the previous revision. A Replacement Narrative may be supplied to describe the justification of the replacement or to provide guidance for implementing the upgrade.

5 Elements that are not revisions of an existing Element must be assigned a Data Object Identifier following current procedures of data administration. The Data Object Revision of new objects begins at 1. If the Element is a revision, the Data Object Identifier is copied from the current version and the Data Object Revision is incremented by 1. All newly created Data Objects must have a Data Object Status of *Work in Process*.

1.5.5 FIELDS

This section summarizes the rules and requirements that apply to the standardized definition of Fields. Where the requirements differ between Standard Fields and Nonstandard Fields, these differences are mentioned.

1 A Standard Field must be a definition of one defined Element. A Nonstandard Field must be based on one Standard Field.

2 Every Field must be created using a particular Data Description Language (DDL). When a DDL is identified, there must be a citation of a source document that contains the syntactical rules for the definition of fields in that DDL.

3 A Standard Field must be present whenever an Element appears on an Optimized Model. There can never be more than one Standard Field for an Element, in a particular DDL.

4 If the conditions that surround the optimization of a particular Data Model prevent the use of the existing Standard Field, then a Nonstandard Field must be defined. The Nonstandard Field must be based on the Standard Field; this means that all the properties of the Standard Field (other that Data Object Identifier, Data Object Revision, Data Object Type, and Data Object Status) are copied into the definition of the Nonstandard Field and these properties are only changed if they must be changed.

5 The Field Datatype must specify a DDL-specific datatype that is consistent with the Data Format of the Element. For Numeric data formats, the specified datatype must also support the Element Scale, Maximum Character Count, and Signed Numeric Indicator of the element. For Text data formats, the specified datatype must support the Maximum Character Count, and the Variable Length Indicator of the element.

6 For Standard Fields, many of the properties must be copied from the Element definition as described below. If the DDL is "CDO" then these properties must be copied directly and generally not modified; but if the DDL is not "CDO" and an Element's definition would violate the syntactic requirements of that DDL for that property, then the property must be modified to meet the DDL's syntactic requirements.

   a. The Field Name is copied from the Element's Abbreviated Name.

   b. The Field Label is copied from the Element's Business Name.

c. The Field Query Name is copied from the Element's Business Name.

d. The Field Edit String is copied from the Element's Element Edit String.

e. The Field Scale is copied from the Element's Element Scale, for Numeric elements.

f. The Field Size is copied from the Element's Maximum Character Count for Text data formats.

g. The Field Justification is copied from the Element's Text Justification, for Text elements.

7 Many of the properties of a Standard Field must be derived from information specified in the element definition, as follows:

a. The Field's Input Validation Expression or Input Validation Routine must implement the rules in the Integrity Narrative of the Element. If an Element has a Code Set defined, the validation of values against this Code Set must be implemented in the Field definition.

b. The Field's Computed-by Expression or Computed-by Routine must implement the rules in the Derivation Narrative of the Element.

c. The Field's Missing Value and/or Initial Value must implement the rules in the Default Action Narrative of the Element.

8 The relationships between this Field and other Fields must be documented as follows:

a. If this Field is composed of some combination of other Fields (sometimes referred to as a Group or Aggregate field), then any ordering rules, special formatting rules, or value constraints must be defined in an Aggregation Narrative.

b. If this Field replaces any other Field, this replacement must be identified. The Data Object Status of the replaced Field must become *Obsolete*. If the Field is a revision of an existing Field, its Data Object Revision must be incremented by one above the previous revision.

9 Fields that are not revisions of an existing Field must be assigned a Data Object Identifier following current procedures of data administration. The Data Object Revision of new objects begins at 1. If the Field is a revision, the Data Object Identifier is copied from the current version and the Data Object Revision is incremented by 1. All newly created Data Objects must have a Data Object Status of *Work in Process*.

1.6 BACKGROUND INFORMATION

1.6.1 CHANGES FROM THE PREVIOUS REVISION

This document contains the results of revising *DEC STD 065-0 Digital Standard Data Definitions Rev C (February 1986)*. That version established the policy and requirements for the definitions of Elements.

Based on the experience gained in developing, reviewing, and implementing standardized element definitions, it was determined that in order to be clearly understood and correctly compared with other element definitions, all elements must be defined and presented within the context of a data model. In addition, it was determined that the definitions of elements and the definitions of fields were properly managed as two separate objects, with an explicit and controlled relationship between them, in order to successfully implement the standardized element definitions within business applications and data bases. Finally, it was determined that the procedures and controls for developing, reviewing, and approving element definitions were too complex and time-consuming, and that the new version of the standard should require a less centralized but still universal means of controlling and integrating definitions.

This version of the standard incorporates those three basic requirements. This version requires the definition of complete, standardized data models that include definitions of entities, relationships, elements, and the data model itself. It provides an explicit distinction and relationship between elements and fields. And it allows any organization to develop and standardize an appropriate data model for their own business that is consistent with (and integratable with) the models from other businesses throughout the corporation, without requiring the approval of a central group.

1.6.2 DEFINITION OF TERMS USED IN THIS DOCUMENT

This section provides definitions of certain terms that are used in this document. Any terms that are the business names of entities or elements names are not defined here; these terms are listed in the table of contents and their definitions appear in subsequent chapters.

Attribute

Attribute is used in this document to refer to any element that is listed as an attribute of an entity or relationship. In this usage, the attributes of an entity are the nonkey elements in that entity. Refer to the definition of the *Has Attribute (Entity)* and *Has Attribute (Complex Relationship)* relationships for further information.

Business Terms Glossary

The Business Terms Glossary (also referred to as the Glossary of Business Terms, or formerly the Keyword Glossary) is a controlled list of the approved Business Terms, including abbreviations and definitions for each term. Refer to the DDNG for more information.

DDNG

The acronym DDNG refers to the *Data Description and Naming Guidelines* document (see Appendix A for ordering information). The DDNG establishes the guidelines for proper descriptions of business data objects. The DDNG also defines the procedure used to transform a proper description into a proper business name, and then to transform this business name into an abbreviated name. The DDNG also describes the guidelines for creating definitions and abbreviations of business terms.

Inheritance

Inheritance is the result of a Classification or Subtyping association between two entities. In these associations, the child entity (that is, the subclass or subtype entity) must have the same primary key and, at a minimum, the same list of attributes as the parent entity. Also, the child entity participates in the same relationships as the parent entity; if participation is mandatory for the parent then it is mandatory for the child, and if it is optional for the parent it may be optional or, if so specified in an Integrity Narrative, mandatory for the child. The child is said to inherit the keys, attributes, and relationships of the parent.

3NF

3NF refers to a level within a defined series of steps for organizing data into entities. The characteristics of data that is in 3NF are as follows.

1 All data items, or elements, must be organized into entities. For each entity, there must be a set of elements (called the primary key) through which every occurrence of that entity can be uniquely identified; the ordering of elements within the entity must be insignificant; a single element may not appear more than once in an entity (note if a single element appears in two or more roles within an entity, these may be identified by separate names); and every element must be atomic (that is, composed of only a single value). For the purposes of compliance with this standard, a complex relationship is considered as an entity in the above definition, in which the primary key is composed of the primary keys of the entities associated.

2 Every nonkey element (that is, every attribute) must be functionally dependent on the complete primary key. Another way of saying this is, that the value of an attribute can be completely determined by knowing the value of the full key; it cannot be sufficient to know just some portion of the full key, nor must it be necessary to know more than the full key.

3 Every nonkey element must depend entirely on the primary key, and on no other attributes; there cannot be transitive dependencies among attributes. Another way of saying this is, the value of each attribute must be able to vary independently of the values of other attributes. For the purposes of compliance with this standard, it is sufficient that no element is algorithmically or deterministically derivable from any other.

1.6.3 DOCUMENT STRUCTURE

The definition of requirements for data object definitions is expressed in this standard as a complete data model, complemented by complete data object definitions of all the entities, relationships, and elements that comprise these models.

Because this document *uses* standard data models to *describe* standard data models, it is in this sense an example of itself. For example, in this document you will see *Element, Relationship, Entity,* and *Data Model* defined as entities, because when the aspects of data modeling are analyzed these are distinct objects of importance to that business. Similarly, you will see Business Name defined as an element, and you will also see the heading **business name** within every data object definition. Every such heading corresponds to one of the elements defined in Chapter 5.

The material in the rest of this standard is organized as follows.

Chapter 2 - Standard Data Models
Chapter 2 contains the 3NF models that define how standardizable data objects can and must be defined and combined to form standard data models. This chapter also includes a section on notations and conventions, which explains the graphic symbols used on the diagrams in this chapter.

Chapter 3 - Standard Entity Definitions
Chapter 3 provides the formal definitions of the entities that appear on the models in Chapter 2. These entities provide the framework through which the information about data objects is organized.

Chapter 4 - Standard Association Definitions
Chapter 4 contains the formal definitions of the associations that are declared on the models in Chapter 2. These definitions provide the rules that govern how the entities can or must be associated with one another on valid standardized models.

Chapter 5 - Standard Element Definitions
Chapter 5 contains the definitions of the individual data elements that are involved in the definition of any standardized data object.

Appendix A - Referenced Documents
Appendix A provides a list of the documents referenced within this standard.

Index
A complete index is provided.

2 STANDARD DATA MODELS

2.1 INTRODUCTION TO THE DATA MODELS

This chapter defines the data models of data objects. Two data models are used to express the definitions of data objects:

- Standard Data Model Requirements
- Type Hierarchy of Data Objects

The first model identifies the data objects that participate on data models, and the required modeling relationships among them. This model expresses, through its cardinalities and supporting documentation, the requirements for 3NF models and optimized models.

The second model shows the structure of a data object definition, which is independent of the modeling context. This model takes advantage of the similarities in form among the various data objects. These similarities are shown using a type hierarchy structure (see Section 2.4 for further information about type hierarchies). The type hierarchy structure allows the model to state, for example, that all *business data objects* have *controllers*, without having to repetitiously state that *3NF entities* have *controllers*, *elements* have *controllers*, and so on.

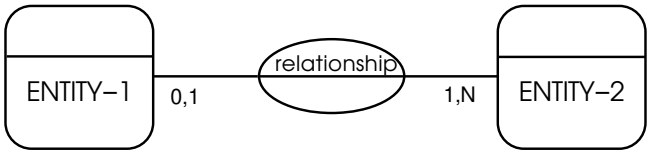
The data model definitions in Section 2.5 consist of textual information, a diagram, and lists of the entities, relationships, and elements that are part of the model. These definitions, like all the data object definitions in this standard, comply with the requirements of this standard.

2.2 SYMBOLS ON THE MODELS

The symbology of the Merise data modeling technique is used for the graphic representation of the models. The two main components in the pictures are entities, which are shown as labeled roundtangles (a rectangle with rounded corners), and relationships, which are shown as labeled ellipses. Connector lines link the entities with the relationships in which they are involved.

An entity represents an object or concept of interest within the business being modeled, and a relationship represents a controlled interaction between two (or more) entities. Entities are identified by their business name. See Chapter 3 for complete, formal definitions of the entities.

Figure 2–1 Sample Modeling Symbols for Relationships



A relationship represents a defined interaction among entities that is of interest to the business. Relationships are identified by their business name, which is the verb or verb phrase that connects the entities. A relationship should be read from left to right (or top to bottom) unless an arrow explicitly indicates the reverse direction. For example, a relationship labeled *Works for* between the entities *Employee* and *Department* should be read, *Employee Works for Department*. See Chapter 4 for complete, formal definitions of the relationships.

When an entity participates in two (or more) relationships, there can be a rule or constraint between the relationships which states that a single occurrence of the entity can participate in *one or the other* of the relationships, but not both. These are also known as *exclusive relationships.* To indicate this on the diagram, an arc labeled with the word "or" is drawn through both connectors. For example, on the *Standard Data Model Requirements* an element may participate in the relationship *has attribute* with a 3NF entity or in the relationship *participates in primary key* with a 3NF entity, but a single element cannot participate in both.

2.3 CARDINALITY

Along the connector lines that join an entity with a relationship, there is a pair of numbers (or a number and the letter N), referred to as the cardinality of the entity within the relationship.

The first number in the pair is the minimum cardinality. The minimum cardinality specifies the minimum number of times each occurrence of that entity can participate in the relationship. If the minimum cardinality is 0, then an occurrence of the entity is not required to participate in this relationship (in some methodologies, this is referred to as *optional* participation). For example, in the relationship *Employee Manages Department*, many of the *Employees* are not managers; so the minimum cardinality of *Employee* within this relationship is 0.

If the minimum cardinality is 1, then every occurrence of the entity must participate in this relationship at least once; this is sometimes referred to as mandatory participation. For example, in the relationship *Employee Works for Department*, every *Employee* must be assigned to a *Department*, so the minimum cardinality is 1.

The second number in the pair is the maximum cardinality. The maximum cardinality specifies the maximum number of times that each occurrence of the entity can participate in this relationship. For example, if the maximum cardinality is 1, then an occurrence of the entity can only participate in the relationship one time. In the relationship *Employee Works for Department*, an *Employee* cannot work for more than one *Department*, so the maximum cardinality is 1.

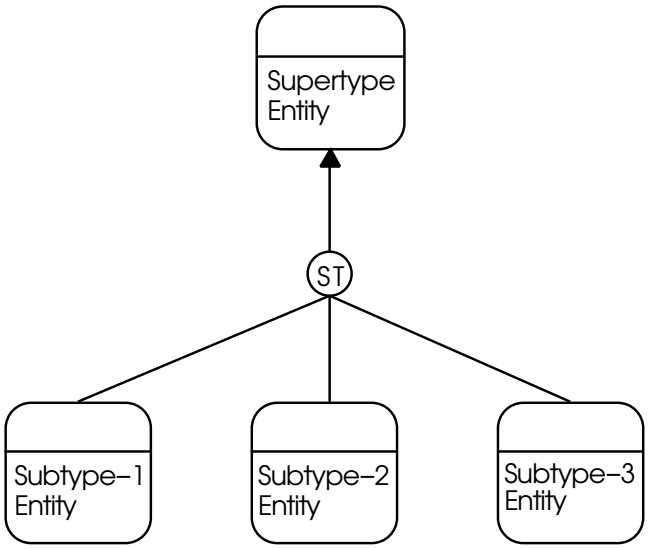
If the maximum cardinality is N, then there is no limit to the number of times the entity can participate in the relationship. For example, in the relationship *Employee Manages Department*, it is possible for one *Employee* to manage more than one *Department*, so the maximum cardinality is N.

The maximum cardinalities of both entities in a relationship can be looked at together. If both maximum cardinalities are 1, then the relationship can be described as a "one-to-one" relationship. If one cardinality is 1 and the other is N, then the relationship can be called a "one-to-many." And if both entities have a maximum cardinality of N, then it is a "many-to-many" relationship.

2.4 SUBTYPES

The model *Type Hierarchy of Data Objects* uses a special form of association called subtyping. Subtyping is indicated by a circle with the letters "ST" inside. See Figure 2–2.

Figure 2–2 Sample Modeling Symbols for Subtyping



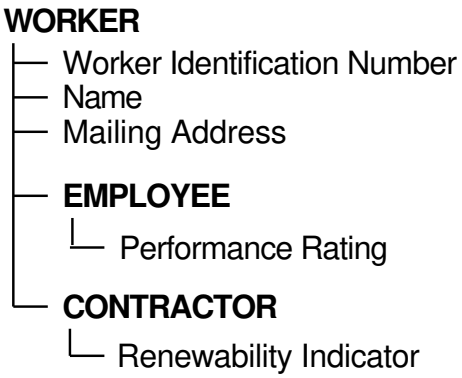
Subtyping expresses the classification of an entity in exhaustive, mutually exclusive partitions. The entity above the "ST" symbol is called the supertype. The entities below the "ST" symbol are called the subtypes.

Every occurrence of the supertype must be an occurrence of only one of the subtypes. For example, the entity *worker* could have the subtypes *employee* and *contractor*. Within this system, every *worker* must be either an *employee* or a *contractor*. Every *contractor*, likewise, is also a *worker*. In a subtyping association, an occurrence of a subtype is at the same time an occurrence of the supertype.

A subtype inherits all of the attributes defined for its supertype. In the previous example, if the entity *worker* has the attribute *worker identification number*, then both *contractor* and *employee* also have this attribute because they are *workers*. Attributes can be assigned to a supertype only if they apply to all the subtypes; individual subtypes do not disinherit any attributes.

All relationships that a supertype participates in are also available for the subtype to participate in. If a *worker* resides in a *state*, then by inheritance *employees* and *contractors* also reside in *states*.

In the attribute lists for the *Type Hierarchy of Data Objects*, the hierarchy of data objects is expressed as an indented list of entities. The attributes that are listed with each entity apply, through inheritance, to all of the entities that are below it on the hierarchy. For example:



In the example shown above, the attributes of *worker* apply universally to all types of workers. If the *worker* is an *employee*, the additional attribute *performance rating* is important; but it does not apply to a *contractor*. Likewise, the attribute *renewability indicator* applies to a *contractor* and not an *employee*.

2.5 DEFINITIONS OF STANDARD DATA MODELS

There are two 3NF models defined in this chapter. The model of *Standard Data Model Requirements* identifies the data objects that participate directly on data models, and the required relationships among them. The model *Type Hierarchy of Data Objects* shows how a data object definition is structured, independent of its participation on a model.

These two models apply simultaneously to the definition of data objects. The first model focuses on how the data objects must be combined; the second model focuses on what they look like individually. All data objects must be an occurrence of one of the subtypes identified on the *Type Hierarchy of Data Objects*; and all data objects must be defined in the context of a model that meets the *Standard Data Model Requirements*.

The two models represent different aspects of the same set of data objects. The criteria by which they are distinguished is as follows: the relationships among data objects that are created on a data model are the subject of the model *Standard Data Model Requirements*; whereas the relationships that are intrinsic to the data object itself, and that do not change from one model to another, are the subject of the model *Type Hierarchy of Data Objects*.

STANDARD DATA MODEL REQUIREMENTS

business name
Standard Data Model Requirements

description
The model of the data objects that appear on standardized data models, showing the requirements and business rules that these models must conform to.

data object type
3NF Model

abbreviated name
STD_DATA_MODEL_REQMT

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The scope of this model is standardized data objects. As defined in this model, every data object that is to be adopted as a standardized data object must be a part of at least one standardized data model, and every standardized data model must consist completely of standardized data objects.

This data model does not include the development processes that are used to develop data models. For example, the identification of the Project for which a model is developed is outside the scope of this standard. Similarly, the standardization procedures that are used to produce and certify standardized data models are outside the scope of this data model. As an example, the identification of the data administrator responsible for maintaining a definition is not shown here. This should not be taken to imply that such information is not important; the reason why it is not included here is that it is not viewed as part of the overall object that is being standardized. The data objects themselves are being standardized, and they do not need to reflect the process or individual who performed the development activity.

This data model reflects the business rules that apply across Digital, regardless of business or geography. Individual business and geographic units may add further requirements beyond those stated here, but they must not reduce these requirements.

notes
This data model focuses on the manner in which standardized data objects can be related on a standardized data model. The further definition of the data objects themselves is the focus of the model, *Type Hierarchy of Data*

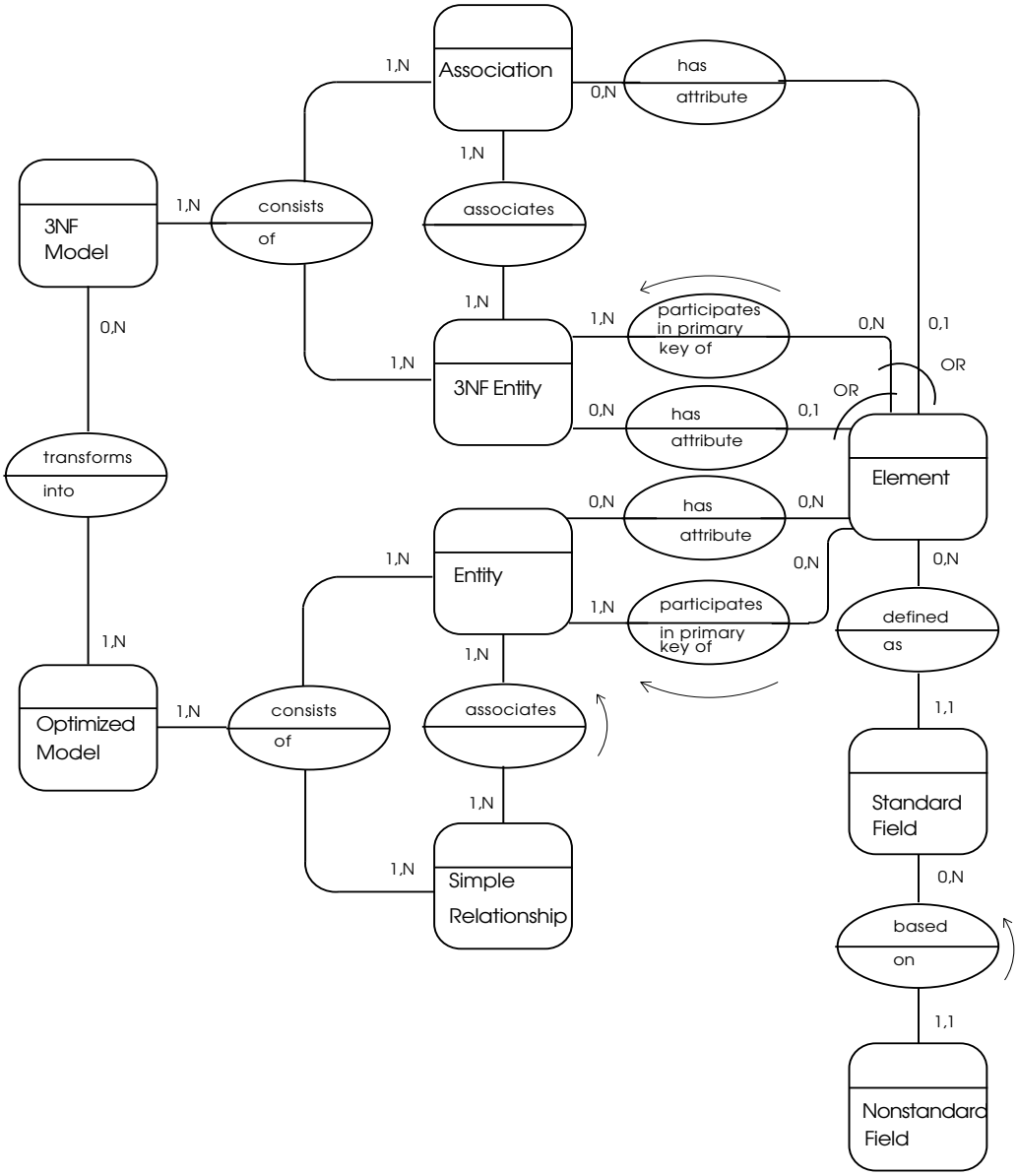
Objects. In order to be completely defined, all data objects must conform to the requirements of both models.

The *Transforms into* relationship that is shown on this model is an occurrence of the *Transforms into* relationship from the *Type Hierarchy of Data Objects*. Its appearance on this model reflects the requirement that each optimized model must be defined as a transformation of one or more 3NF models.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

Figure 2–3 Standard Data Model Requirements



entities, keys, and attributes

The following list identifies the entities on the *Standard Data Models Requirements* model, and lists the keys and attributes of each entity. The symbol *[K]* identifies the elements that participate in the primary key.

3NF ENTITY

Abbreviated Name
Business Name
Data Object Identifier *[K]*
Data Object Revision *[K]*
Data Object Status
Data Object Type
Description
Integrity Narrative
Notes

3NF MODEL

Abbreviated Name
Business Name
Data Object Identifier *[K]*
Data Object Revision *[K]*
Data Object Status
Data Object Type
Description
Integrity Narrative
Notes

ASSOCIATION

Abbreviated Name
Business Name
Data Object Identifier *[K]*
Data Object Revision *[K]*
Data Object Status
Data Object Type
Description
Integrity Narrative
Notes

ELEMENT

Abbreviated Name
Business Name
Data Object Identifier *[K]*
Data Object Revision *[K]*
Data Object Status
Data Object Type
Default Action Narrative
Derivation Narrative
Description
Integrity Narrative
Notes

ENTITY

Abbreviated Name
Business Name
Data Object Identifier *[K]*
Data Object Revision *[K]*
Data Object Status
Data Object Type
Description
Integrity Narrative
Notes

NONSTANDARD FIELD

Computed-by Expression
Computed-by Routine Name
Data Object Identifier *[K]*
Data Object Revision *[K]*
Data Object Status
Data Object Type
Field Datatype
Field Edit String
Field Help
Field Justification
Field Label
Field Name
Field Query Name
Field Scale
Field Size
Initial Value
Input Validation Expression
Input Validation Routine Name
Missing Value

OPTIMIZED MODEL

Abbreviated Name
Business Name
Data Object Identifier *[K]*
Data Object Revision *[K]*
Data Object Status
Data Object Type
Description
Integrity Narrative
Notes

SIMPLE RELATIONSHIP

Abbreviated Name
Business Name
Data Object Identifier *[K]*
Data Object Revision *[K]*
Data Object Status
Data Object Type

Description
Integrity Narrative
Notes

STANDARD FIELD

Computed-by Expression
Computed-by Routine Name
Data Object Identifier *[K]*
Data Object Revision *[K]*
Data Object Status
Data Object Type
Field Datatype
Field Edit String
Field Help
Field Justification
Field Label
Field Name
Field Query Name
Field Scale
Field Size
Initial Value
Input Validation Expression
Input Validation Routine Name
Missing Value

relationships and attributes

The relationships on the model *Standard Data Model Requirements* are listed below, along with any attributes of these relationships.

ASSOCIATES
Minimum Cardinality
Maximum Cardinality
BASED ON
CONSISTS OF (Data Model)
DEFINED AS
HAS ATTRIBUTE (Complex Relationship)
HAS ATTRIBUTE (Entity)
PARTICIPATES IN PRIMARY KEY OF
TRANSFORMS INTO
Transformation Narrative

TYPE HIERARCHY OF DATA OBJECTS

business name
Type Hierarchy of Data Objects

description
The model of data objects that shows explicitly how data objects are classified, and defines what information is needed to define each type of data object.

data object type
3NF Model

abbreviated name
TYPE_HRCHY_DATA_OBJCT

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The scope of this model includes all data objects that have standardized data definitions, together with other objects that can or must be part of these standardized data definitions.

This model does not include the possible relationships among data objects of different types. The required relationships among data objects (for example, between *element* and *field*) are the focus of the model, *Standard Data Model Requirements*. Data objects on this model are shown independent of how they appear on actual standardized models.

This data model does not include the development processes or standardization procedures that are used to produce and certify standardized data models.

This data model reflects the business rules that apply across Digital, regardless of business or geography. Individual business and geographic units may add further requirements beyond those stated here, but they must not reduce these requirements.

There are additional data objects that at this time cannot be standardized; for example, Records. In addition, there are other object types such as Program Units that are not data objects but which are relevant to the broader context of modeling, design, development and control of business systems. Future versions of this model may be extended to cover some of these other data objects and other objects, but according to the current model these items cannot have standardized definitions.

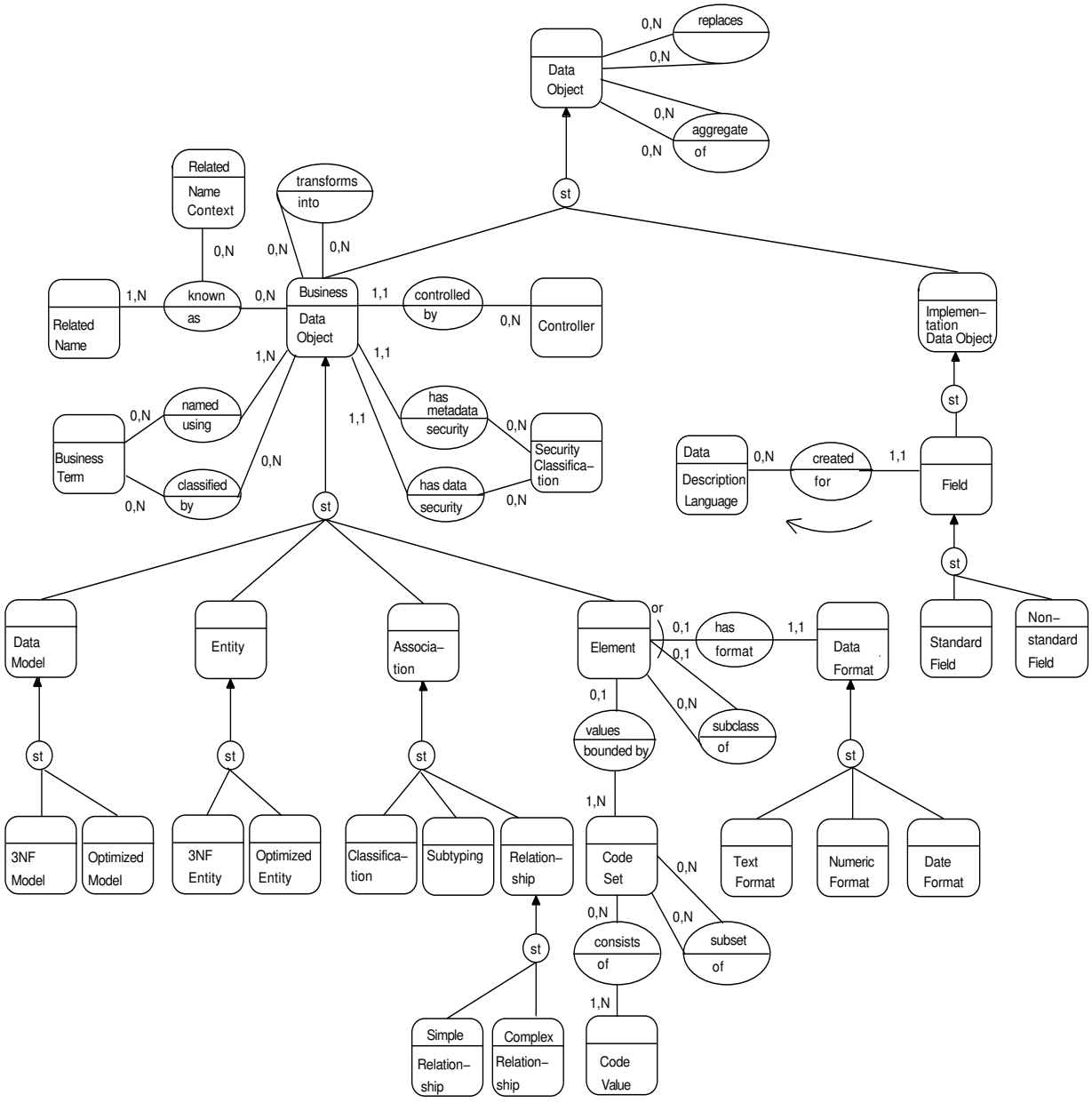
notes

The *Type Hierarchy of Data Objects* focuses on the semantic relationships among the various types of data objects. This model conveys information such as, "All *business data objects* must have a *controller*." This rule applies whether the data object is a *3NF model* or a *simple relationship*, or some other type of data object. Rather than declare the rule explicitly for every type of *data object*, the concepts of subtyping and inheritance are employed to simplify the model. Refer to subhead 2.4 for further information on how to interpret this aspect of the model.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

Figure 2–4 Type Hierarchy of Data Objects



entities, keys, and attributes

The lists on this and the following page define the attributes of each entity on the *Type Hierarchy of Data Objects*. Wherever an entity appears below another entity on the hierarchy, the lower-level entity is indented. All of the attributes of the higher-level entity are inherited by all entities that are below it hierarchically. The symbol *[K]* identifies the elements that participate in the primary key.

DATA OBJECT
- Data Object Identifier *[K]*
- Data Object Revision *[K]*
- Data Object Type
- Data Object Status
- **BUSINESS DATA OBJECT**
  - Business Name
  - Description
  - Abbreviated Name
  - Integrity Narrative
  - Notes
  - **DATA MODEL**
    - **3NF MODEL**
    - **OPTIMIZED MODEL**
  - **ENTITY**
    - **3NF ENTITY**
    - **OPTIMIZED ENTITY**
  - **ASSOCIATION**
    - **SUBTYPING**
    - **CLASSIFICATION**
    - **RELATIONSHIP**
      - Reverse Relationship Name
      - **SIMPLE RELATIONSHIP**
      - **COMPLEX RELATIONSHIP**
  - **ELEMENT**
    - Derivation Narrative
    - Default Action Narrative
- **IMPLEMENTATION DATA OBJECT**
  - **FIELD**
    - Field Name
    - Field Help
    - Field Datatype
    - Field Label
    - Field Size
    - Field Query Name
    - Field Scale
    - Field Edit String
    - Field Justification
    - Input Validation Expression
    - Input Validation Routine Name
    - Computed–by Expression
    - Computed–by Routine Name
    - Missing Value
    - Initial Value
    - **STANDARD FIELD**
    - **NONSTANDARD FIELD**

DATA FORMAT
- Data Format Type *[K]*
- Element Edit String
- Maximum Character Count
- **TEXT FORMAT**
  - Variable Size Indicator
  - Text Justification
- **NUMERIC FORMAT**
  - Element Scale
  - Signed Numeric Indicator
- **DATE FORMAT**
  - Relative Date Indicator

The preceding entities are those that appear directly within the hierarchical structure in this model. The other entities that appear in this model are listed, with their attributes, below.

BUSINESS TERM
Business Term Abbreviation *[K]*
Business Term *[K]*
Business Term Definition
Value Type Indicator
CODE SET
Code Set Name *[K]*
CODE VALUE
Code Value *[K]*
Code Description
CONTROLLER
Controller Name *[K]*
DATA DESCRIPTION LANGUAGE
Data Description Language Name *[K]*
Documentation Source Citation
RELATED NAME
Related Name *[K]*
RELATED NAME CONTEXT
Related Name Context *[K]*
SECURITY CLASSIFICATION
Security Classification Name *[K]*

relationships and attributes

The relationships in the *Type Hierarchy of Data Objects* are listed below, along with any attributes of these relationships.

AGGREGATE OF
Aggregation Narrative
CLASSIFIED BY
CONSISTS OF (Code Set)
CONTROLLED BY
Controller Contact Employee Identifier
CREATED FOR
HAS FORMAT
HAS DATA SECURITY
HAS METADATA SECURITY
KNOWN AS
NAMED USING
REPLACES
Replacement Narrative
SUBCLASS OF
SUBSET OF
TRANSFORMS INTO
Transformation Narrative
VALUES BOUNDED BY

3 STANDARD ENTITY DEFINITIONS

This chapter contains the definitions of each of the entities that appear on the two models in Chapter 2, *Standard Data Model Requirements*, and *Type Hierarchy of Data Objects*.

Each of these entities represents some item or category of information that is related to the complete definition of a data object.

3NF ENTITY

business name
3NF Entity

attributes

Data Object Identifier *[K]*
Data Object Revision *[K]*
Data Object Type
Data Object Status
Business Name
Description
Abbreviated Name
Integrity Narrative
Notes

description
An object about which a business wishes to collect information, that conforms to the requirements of the third form of normalization.

data object type
3NF Entity

abbreviated name
3NF_ENTY

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Every 3NF entity must appear on at least one 3NF model. A 3NF entity must be evaluated for normalization within the context of the 3NF model(s) it appears on.

The requirements of the third normal form are restated in virtually every methodological guidebook of data modeling. In summary, the requirements are:

1. All data is organized into entities that have primary keys.
2. No attribute can appear more than once in any entity; there cannot be repeating groups.
3. Every attribute must depend on the complete primary key; there cannot be dependence on just a portion of the key.
4. Every attribute must depend entirely on the key, and on no other attributes; there cannot be transitive dependencies among attributes.

An element cannot appear as an attribute of more than one 3NF entity, unless the attribute is inherited through Subtyping or Classification. This requirement spans all standardized 3NF entities. That is, if an element appears as a nonkey attribute in one 3NF entity on any 3NF model, it cannot appear as an attribute of any other 3NF entity.

3NF Entity is a subtype of *Entity*.

notes

To be in third normal form, an entity must meet certain formal requirements. The steps of normalization can be found in virtually every textbook or methodology definition that deals with data analysis, data modeling, or data base design. These requirements were originally specified by E.F. Codd in the paper, "Further Normalization of the Data Base Relational Model" (in *Data Base Systems*, Courant Computer Science Symposia Series, Vol. 6, Prentice-Hall, 1972).

Normalization of data is essentially a means for determining the fundamental types of entities that are of interest to the enterprise, and for determining for each of these what type of information is to be recorded. Normalization ensures that contradictory values for the same piece of data do not occur.

3NF entities may appear on optimized models as well as on 3NF models. 3NF entities can be the foundation for the definition of implemented data structures. Typically, an implementation that is based directly on 3NF entities is optimized in favor of flexibility and integrity, not in favor of performance.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

3NF MODEL

business name
3NF Model

attributes

Data Object Identifier *[K]*
Data Object Revision *[K]*
Data Object Type
Data Object Status
Business Name
Description
Abbreviated Name
Integrity Narrative
Notes

description
An entity-relationship data model in which all the entities are in third normal form.

data object type
3NF Entity

abbreviated name
3NF_MODEL

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The requirements for 3NF models are expressed in the model, *Standard Data Model Requirements*. All entities on a 3NF model must be 3NF entities. See the definition of *3NF Entity* in this chapter, for the definition of third normal form. A 3NF model may contain any of the types of associations: Complex Relationship, Simple Relationship, Subtyping, or Classification.

Every 3NF model must be supported by an Entity-Relationship Diagram that shows (at a minimum) all the entities and relationships in the model. The entities and relationships must be identified using their Business Name. The diagram can use any graphical modeling symbols or notation. For example, "Crow's Feet" may be used to indicate cardinality; diamonds may be used to denote relationships; and so on.

3NF Model is a subtype of *Data Model*.

notes
A 3NF model can be implemented directly; more commonly, however, it is transformed into an optimized model before implementation.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

related names

related name	context
Conceptual Model	Merise
Entity-Relation Model	Common Usage
Entity-Attribute-Relation Model	Common Usage
Extended Entity-Relation Diagram	DECdesign
Application Logical Data Model	Data Management Handbook (Information Services Europe)

ASSOCIATION

business name
Association

attributes

Data Object Identifier *[K]*	Business Name
Data Object Revision *[K]*	Description
Data Object Type	Abbreviated Name
Data Object Status	Integrity Narrative
	Notes

description
A defined association between entities on a data model.

data object type
3NF Entity

abbreviated name
ASSCTN

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Every association must appear on a model. Without the context of a model, the definition of the association is not complete. Note also that an association can appear on more than one model.

Every association must connect to at least two entities, unless it is a *recursive* relationship. A recursive relationship is one that connects an entity with itself. Subtyping and Classification cannot be recursive.

When an association appears on a model, all the entities it associates must also appear on that model. In some modeling techniques, an association can be linked to an entity that appears to be off the model; such entities are sometimes called *external entities*. However, for the purposes of evaluating the model for normalization and standardization, these external entities are treated as part of the model.

Association is a subtype of *Business Data Object*.

notes
An association is any controlled connection among entities on a data model. There are three types of association: Classification, Subtyping, and Relationships.

Classification and Subtyping are ways of referring to some specialized grouping within an entity. Subtyping is a means for strictly categorizing every occurrence of an entity into one of a set of mutually exclusive types. Classification is a means for referring to any selectable subset of the occurrences of an entity. Subtyping is typically defined by reference to the value of one attribute of the entity, whereas Classification is often defined by virtue of a relationship that the occurrence of the entity participates in.

Both Classification and Subtyping provide a means to redefine an entity into various, more specialized entities. In both of these types of association, the specialized entity must have the same primary key, and at least the same list of attributes (additional attributes may be added to the list, if they apply to the specialized entity). This passing down of attributes from the general entity to the specialized one is called *inheritance*. See the definitions of *Subtyping* and *Classification* in this chapter for additional information.

Relationships, on the other hand, are connections representing a particular fact about the business being modeled; specifically a fact about the connection between distinct objects.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

BUSINESS DATA OBJECT

business name
Business Data Object

attributes

Data Object Identifier *[K]*	Business Name
Data Object Revision *[K]*	Description
Data Object Type	Abbreviated Name
Data Object Status	Integrity Narrative
	Notes

description
An item or grouping of information that is significant to a business. A Business Data Object definition provides all the information that a person needs to know about that data object to use it within their business.

data object type
3NF Entity

abbreviated name
BUSNS_DATA_OBJCT

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
All Business Data Objects must be named in accordance with the DDNG.

All Business Data Objects must have a controller who is responsible for the accuracy and usefulness of the definition.

All Business Data Objects must be given two security classifications: one for the definition, and one for the occurrences of that data object.

Business Data Object is a subtype of *Data Object*.

notes
A business data object refers to something about the business, not something about the information system that supports the business.

Business Data Objects can often be identified by looking for the following:

a. Tangible things—A singular item that can be seen and touched, such as a person, a location, or a piece of equipment.

b. Conceptual things—An item that cannot be touched, but which serves to organize or implement a business, such as an account, a marketing program, or a project.

c. Roles—When one object can be viewed in multiple roles. For example an employee can be a supervisor, or a doctor can be a patient; the role played is different than the object that is playing the role, so the role itself is an object.

d. Incidents—An occurrence or event that happens at a specific time, such as a service call, system crash, accident, or shipment.

e. Interactions or Relationships—A transaction, contract, or business rule that relates two or more objects in the business. For example: a customer *initiates* an order; an employee *is located at* an office; a payment *is credited to* an account.

f. Qualities, Characteristics, or Categories—The factors in business decisions or in controlling operations; facts about an object. For example: address, price, category name.

g. Specifications—Representation of standards or tables. For example: part category table, chart of accounts.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

related names

related name	context
Standard Data Definition	Data Administration

BUSINESS TERM

business name
Business Term

attributes

Business Term *[K]*
Business Term Abbreviation *[K]*
Business Term Definition
Value-Type Indicator

description
A term, used within a business, that is formally defined and uniquely identified.

data object type
3NF Entity

abbreviated name
BUSNS_TERM

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
An abbreviation can have only one definition, and a single definition cannot apply to more than one abbreviation. Two different business terms may have the same abbreviation, if they have exactly the same meaning (synonyms). For example, "confirm" and "verify" can mean the exact same thing, so only one abbreviation (for example, "CNFRM") should be used for to refer to this meaning. On the other hand, a single business term may have two different meanings (homonyms), in which case it must have two different abbreviations. For example, "Second" can mean "Unit of time; 1/60th of a minute", or it can mean "Next after the first." In each case, the term is given a distinct abbreviation (for example, SCND and SECOND) in order to avoid the ambiguity.

Note that current practice requires a manual quality control review and approval process before an entry can be added to the Business Term Glossary. Refer to the DDNG for information on the creation of new entries in the Business Term Glossary, specifically for the integrity rules regarding homonyms and synonyms among business terms.

notes
The requirements and process for creating standard business terms and approved definitions are defined in the DDNG.

Some business terms represent the kind of data that is being named, such as *Identifier, Amount, Count*, or *Name*. These business terms are called *value types*. All names of *elements* should end in a value type. Refer to the DDNG for further information.

security constraints

subject	classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

related names

related name	context
Keyword	Data Administration
Subject Area	Data Administration
Data Subject	Data Administration

CLASSIFICATION

business name
Classification

attributes

Data Object Identifier *[K]*
Data Object Revision *[K]*
Data Object Type
Data Object Status
Business Name
Description
Abbreviated Name
Integrity Narrative
Notes

description
A classification of an entity into categories that have additional attributes or additional business rules which apply to them.

data object type
3NF Entity

abbreviated name
CLSFTN

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Classification establishes a hierarchy between a parent entity and one or more classes of that parent entity, which are called the child or the subclass entities. The child entity is a redefinition of the parent, which adds new attributes or business rules, but which continues to conform to the definition of the parent entity.

The subclass entity must inherit all the keys and attributes of the superclass, as well as all the relationships in which the superclass participates. The subclass entity may also have additional attributes and participate in additional relationships that the superclass does not.

The criteria by which a parent is classified into its subclasses must be defined in the integrity narrative of the association's definition.

Classification cannot appear on an optimized model, because such an optimization would require technology that intrinsically supports inheritance. When the use of such technologies becomes standardizable, this restriction may be removed.

Classification is a subtype of *Association*.

notes

Classification provides a means for defining specific categories into which an entity is divided. These categories are themselves defined as entities, and as such can participate in relationships in data models.

Classification is often thought of as a KIND-OF relationship. Classification allows for simultaneous use of the superclass and the category entities on 3NF models. Without the benefit of Classification, these simultaneous definitions would be invalid because it would result in two or more 3NF entities that have the same primary key. For example, the superclass entity Trading Partner can be categorized into the entities Supplier and Customer; subsequently, the entity Supplier can appear on models where it is of particular interest to the business.

Classification could be transformed in a variety of ways when a model is optimized. For example, the superclass can be carried down to the optimized model intact, where it serves as the vehicle for the core attributes and relationships. Meanwhile the subclasses are carried down as optimized entities, discarding the nonkey attributes of the superclass; the categories are then each linked back to the supertype using a series of explicit simple relationships. Another optimization approach is to transform all the subclasses into a single optimized entity, which carries the complete set of attributes from all of the categories including the inherited ones. With this approach there would be no need for an optimized entity corresponding to the superclass.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

related names

related name	context
Inheritance	Object-Oriented Systems

CODE SET

business name
Code Set

attributes

Code Set Name *[K]*

description
The collection of code values that apply to the same element.

data object type
3NF Entity

abbreviated name
CD_SET

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The values in the code set of an element are controlled by the controller of that element.

When one element is a subclass of another, its code set must be a subset of the parent element's set of valid values.

notes
A code set is the complete set of values that an element can have.

A code set is normally created for any element that has a value type of *Code* or *Indicator*. A code set must not be created for elements that are *Identifiers*. For example, an element *Transfer Reason Code* should have a code set, but the element *Center of Control* must not. Code sets should only be used when the element contains an abbreviated form of the actual value, or when the actual values must occur on a limited list of legal values. A code set must not be used if the number of possible values is large (a reasonable limit would be fifty values, although exceptions are possible), or if the set of values changes frequently (no more than an average of one change to the table per month, although again exceptions are possible).

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

CODE VALUE

business name
Code Value

attributes

Code Value *[K]*
Code Description

description
A code or official abbreviation for a particular data value.

data object type
3NF Entity

abbreviated name
CD_VAL

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The controller of the data element is the only source that is authorized to add new code values or alter existing ones.

Code Value is a weak entity, that depends on its participation in a Code Set for its identification. In other words, the meaning of an individual code value cannot be determined without knowing on which code set it must be evaluated.

notes
A code value is a single code or abbreviation for a single data value.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

related names

related name	context
Valid Values	Datatrieve Syntax

COMPLEX RELATIONSHIP

business name
Complex Relationship

attributes

Data Object Identifier *[K]*
Data Object Revision *[K]*
Data Object Type
Data Object Status
Business Name
Description
Abbreviated Name
Reverse Relationship Name
Integrity Narrative
Notes

description
A relationship that involves some complex structure: either it involves more than two entities, has attributes, or involves a many-to-many cardinality.

data object type
3NF Entity

abbreviated name
CMPLX_RLNSP

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
A complex relationship is any relationship among entities which either:

a. Relates more than two entities

b. Expresses a many-to-many maximum cardinality, or

c. Has fully dependent attributes

Some modeling methodologies do not permit complex relationships on 3NF models. This standard does not impose that requirement, but it may be established within operating groups as a local requirement.

A complex relationship cannot appear on an optimized model.

The following requirements must be met for any complex relationship that has attributes:

- The attributes must be evaluated for normalization as if they appeared on an entity whose primary key was the concatenation of all the keys of the entities joined by this complex relationship.

- These attributes can appear on some other 3NF model under a 3NF entity; the key of this 3NF entity must be the concatenation of the keys of all entities joined by this complex relationship. Essentially, this allows for the parallel definition of a 3NF entity alongside a complex relationship, for use on models where the technique prohibits complex relationships. These parallel objects must be connected by a "transforms into" relationship.
- The attributes cannot appear on any other complex relationship, on any other model.

Complex Relationship is a subtype of *Relationship*.

notes

A complex relationship is a relationship that either joins more than two entities, involves a many-to-many maximum cardinality, or has attributes.

An example of joining more than two entities would be the relationship "Provides Support To" between an Employee and a Customer. Many employees can deliver various types of service to a customer, and the service type itself can be modeled as a third entity that participates in the relationship.

An example of a many-to-many relationship would be the relationship "Works On" between an Employee and a Project. An employee can work on many projects; and a project can have many employees that work on it. Because both of the maximum cardinalities are "N" for this relationship, it becomes a complex relationship.

An example of a relationship that has attributes would be the relationship "Works For" between an Employee and a Department. The attributes of this relationship could include "Start Date" and "Temporary Assignment Indicator", or any other attribute that is not directly about the employee nor the department, but rather is a fact about the relationship between them.

A complex relationship usually is transformed into an entity on an optimized model prior to being implemented. For example, the 3NF model *Standard Data Model Requirements*, defined in Chapter 2 of this standard, contains the complex relationship *Has Attribute* that links the entities *Entity* and *Element*. This is a complex relationship because both entities have a maximum cardinality of *N*; that is, it is a many-to-many relationship. On an optimized model, this relationship would be transformed into an optimized entity, perhaps called *Attribute*. This new entity would have a compound key, consisting of the key of the Entity and the key of the Element, and would be joined using simple relationships to these associated entities. If the original complex relationship had had any attributes, these would become attributes of the new entity. This is one form of Transformation: the original complex relationship has been transformed into an entity and two simple relationships.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

related names

related name	context
Junction Record	Data Base Design

CONTROLLER

business name
Controller

attributes

Controller Name *[K]*
Controller Contact Employee Identifier

description
An agent or agency of the corporation that has the knowledge, authority, and responsibility to ensure the accuracy and usability of the definition of a business data object.

data object type
3NF Entity

abbreviated name
CNTRLR

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The controller must be the agent or agency that provides the greatest span of authority, responsibility, and knowledge within the enterprise for the particular business data object they control. A controller can be an organization, a standing committee or staff, or an individual acting in a particular role, depending on how the requirement for authority, responsibility, and knowledge is best fulfilled. Refer to Section 1.4.2 for further explanation of the responsibilities of a controller.

The expected permanence of the controller should be roughly the same as the expected permanence of the business object being defined. A project or project team should not be a controller, because projects by definition come to an end when their objectives are met. If the project will include a role that will continue after the project has finished (for example, a business architect role), then this role can become the controller.

notes
If the controller is an individual, it must be identified by an organizational role title or job title rather than by personal name. Controllership is associated with the position of responsibility and authority, rather than an individual personality. If the individual moves to a new organization, the controllership does not move with him or her.

For some cross-organizational or cross-geographic domains where no individual or organization has the authority to control the others, the controller is best identified as the collective management of that domain (for instance, a "board of directors" or an "area manager's forum" specific to that domain).

For each data definition, the controller may identify a specific employee who becomes the contact point for review, feedback, questions, and resolution of conflict relative to the definition of a business data object.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

DATA DESCRIPTION LANGUAGE

business name
Data Description Language

attributes

Data Description Language Name *[K]*
Documentation Source Citation

description
A computer language that defines data and data structures.

data object type
3NF Entity

abbreviated name
DDL

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Data Description Languages (DDLs) sometimes fall into families. For example, VAX CDD/Plus provides the language known as CDO (Common Dictionary Operator), which is capable of recording data definitions that can be used by other VAX languages. Whenever an existing DDL such as CDO can accommodate the needs of another software product, the other product should not be identified as a DDL.

notes
DDLs are the computer languages that define data and data structures. In most cases, higher level languages such as BASIC or COBOL have the capability of defining data. However, any language that has a supported interface to the CDD/Plus data dictionary can utilize definitions written with the Common Data Dictionary's (CDD) Common Dictionary Operator (CDO) utility, using the CDO language. Thus the DDL called CDO is the most generic language for use within Digital.

Third-party query tools and data base managers typically provide their own DDL. If these third-party tools do not support an interface to CDD /Plus, then these DDLs are valid for defining data and data structures.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

DATA FORMAT

business name
Data Format

attributes

Data Format Type *[K]*
Element Edit String
Maximum Character Count

description
The specification of the format of a data element.

data object type
3NF Entity

abbreviated name
DATA_FRMT

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Data Format is a weak entity; it depends on its connection to an element for its identity.

The data format type for a standardized element must be defined as either *Text, Numeric,* or *Date*. Although other data formats exist (for example, graphics or voice data), elements of these formats cannot be standardized in this version.

notes
The data format of an element contains the definition of the element's standardized physical properties: its length, the rules for its presentation on video or hardcopy reports, and so on. The physical properties that are needed for an element vary, depending on the type of data represented by the element: *Text, Numeric*, or *Date*. Refer to the entity definitions of the entities Text Format, Numeric Format, and Date Format for the list of unique properties that are applicable to each form of data.

Numeric is used if the data represents an item that can be an operand in an arithmetic expression, such as addition or averaging (for example, "Quantity Ordered" or "Days Sales Outstanding").

Date is used if the data represents a point in time, whether it is a fiscal date, calendar date, time of day, or combination of date and time. *Date* is also used for any block of time, such as a month. Note that a *nonstandard field* must be defined whenever there is a need to represent a point in time in a data format other than *Date*.

Text is used in all other cases. *Text* allows for definition of fixed- or variable-length data items.

A Data Format belongs to a single element (referred to as the root element), but is shared by all elements that are a subclass of that root element. This sharing can propagate, if the subclass elements in turn have their own subclasses. Refer to the definition of *Subclass of* in Chapter 4 for information and restrictions on the use of subclasses. Note in particular that two elements do *not* share a Data Format merely because they look the same; they *must* participate in a subclass relationship with a common root element in order to share a Data Format.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

DATA MODEL

business name
Data Model

attributes

Data Object Identifier *[K]*	Business Name
Data Object Revision *[K]*	Description
Data Object Type	Abbreviated Name
Data Object Status	Integrity Narrative
	Notes

description
A formal description of the business information and its interrelationships. A Data Model has boundaries within which the model is complete.

A Data Model includes both the schematic, pictorial representation of the business information and the complete definitions of the data objects shown on that schematic.

data object type
3NF Entity

abbreviated name
DATA_MODEL

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The model *Standard Data Models Requirements*, defines the minimum requirements for the definition of data models and the component parts of models.

The scope of a data model must be defined in the *integrity narrative* of the standard definition of that model. The scope of a data model must define the boundaries of the model in terms of the business. Where appropriate, the scope should indicate possible future extensions of the boundaries.

Data Model is a subtype of *Business Data Object*.

notes
Different modeling techniques define different kinds of data models. However, across all techniques there are certain commonalities. First, all techniques include the step of normalizing the data model. Also, all techniques include a step wherein the normalized model is altered in support of implementation concerns such as access time and data base technology. These two types of models are known respectively as *3NF*

Models and *Optimized Models* in the model *Type Hierarchy of Data Objects*. Other types of data models cannot be standardized under the current version of this standard, and are outside the scope of this definition.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

DATA OBJECT

business name
Data Object

attributes

Data Object Identifier *[K]*
Data Object Revision *[K]*
Data Object Type
Data Object Status

description
A description or definition of data.

data object type
3NF Entity

abbreviated name
DATA_OBJCT

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
No data object can be defined in isolation from other data objects. All data objects must be defined in the context of a data model. Although the definition of a data object must be independent of the models it appears in, the definition cannot be valid if the object does not appear on any model.

notes
Any controlled set of information about data is a data object. This includes data models, field definitions, and all other types of data object as shown in the model, *Type Hierarchy of Data Objects*.

Although most data objects have names, the integrity requirements and form of these names will vary from one data object type to another. In casual or local practice, the name of the data object may serve as the primary means of identifying it (in dictionaries and other data management tools, an abbreviated name will often serve as an identifier). However, the model and definitions in this standard do not require the name of a data object to be unique; for example, there can be one or more standardized optimized entities called "Employee" among all the optimized models in the corporation. As a result of these considerations, the primary key of a data object, from an enterprise-wide perspective, is not the name. Instead, a nonsignificant identifier is assigned at the time an object is created that serves to differentiate and identify all data objects in all contexts.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

related names

related name	context
Standard Data Definition	Data Administration
Dictionary Entry	DECdesign

DATE FORMAT

business name
Date Format

attributes

Data Format Type *[K]*
Element Edit String
Maximum Character Count
Relative Date Indicator

description
The format of a data element that represents a point in time.

data object type
3NF Entity

abbreviated name
DT_FRMT

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The Date Format must be employed for any data element that represents a point in time or an elapsed time period. This includes calendar dates, time of day, or a combination of date and time.

Date Format is a subtype of the entity *Data Format*. It is a weak entity that depends entirely on the element to which it is connected for its identification.

notes
A Date Format implies that the datatype of the fields that define this element will be in VMS-date format, if the host DDL supports it (other DDLs will use their proprietary date-storage format as appropriate).

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

ELEMENT

business name
Element

attributes

Data Object Identifier *[K]*
Data Object Revision *[K]*
Data Object Type
Data Object Status

Business Name
Description
Abbreviated Name
Integrity Narrative
Notes
Derivation Narrative
Default Action Narrative

description
An individual unit of data that has meaning within a business context; a simple fact about an object.

data object type
3NF Entity

abbreviated name
ELEMT

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
An element definition must provide all the information that is needed for a person to understand or implement the data item.

The definition of any element that is computed or derived must include a *derivation narrative*. Derived elements cannot appear on 3NF entities, in accordance with the rules of normalization.

An element definition cannot be standardized outside the context of an approved Entity definition.

An element must have a Data Format defined for it, unless it is a subclass of another element (in which case it shares the data format of its root element).

Element is a subtype of *Business Data Object*.

notes
An element is a fact, quality, characteristic, or category that identifies or describes a business object or event.

Elements and fields define data items. Element definitions provide the information needed by people to understand and control the data item, while field definitions provide the information needed by computers to store or process the data item.

An element can be an atomic or group item, or it can be derived from other elements. An atomic data item is one that cannot be meaningfully decomposed within its business context (for example, Employee Number). A group item is one that is an aggregate of other elements that, when grouped together, are consistently identified by a single name (for example, Address). A derived element is one in which values are computed from the values of other elements (for example, Account Balance Amount).

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

related names

related name	context
Attribute	Data Structures
Segment	Transaction Standards

ENTITY

business name
Entity

attributes

Data Object Identifier *[K]*
Data Object Revision *[K]*
Data Object Type
Data Object Status
Business Name
Description
Abbreviated Name
Integrity Narrative
Notes

description
A class of persons, places, things, or business concepts about which the business wishes to collect data.

data object type
3NF Entity

abbreviated name
ENTY

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
A definition of an entity must always define the entity as an independent business object, rather than defining the role that the entity plays within the model. The definition of an entity on a particular model must be written in a manner that allows the entity to be incorporated into different models.

An Entity cannot be standardized outside the context of an approved data model.

Entity is a subtype of *Business Data Object*.

notes
Entities are the principle vehicles for organizing business information and are the objects that the business data refers to.

There are two types of entities that can be standardized in accordance with this standard: *3NF Entities* and *Optimized Entities*. Refer to the definitions of these objects in this chapter for further information.

Entities are a design/analysis construct that typically are ultimately implemented as records or relations (depending on the technology used). Entities represent any collection of elements that has a primary key. The business content of the entity should not be altered when it is implemented; if an alteration is needed, the entity definition should be changed.

Some modeling methodologies distinguish between regular entities and "weak entities." A weak entity is one that does not have a primary key that uniquely identifies one occurrence from another; weak entities rely on some other entity to be identified first, after which the occurrence of the weak entity can be identified. The most common example of this is the entity "Dependent" in a personnel model. "Dependents" can only be identified by first identifying the employee that they depend on. This standard does not support a unique category of "weak entities"; they must be defined as either a 3NF entity or an optimized entity, and their Integrity Narrative must explain their dependence on another entity.

security constraints

subject	classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

FIELD

business name
Field

attributes

Data Object Identifier *[K]*
Data Object Revision *[K]*
Data Object Type
Data Object Status
Field Name
Field Label
Field Help
Field Datatype
Field Size
Field Query Name
Field Scale
Field Edit String
Field Justification
Input Validation Expression
Input Validation Routine
Computed-by Expression
Computed-by Routine
Missing Value
Initial Value

description
The fundamental identifiable unit of data for computer processing required to design and to run an application.

data object type
3NF Entity

abbreviated name
FLD

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
A field definition must provide all the information that is needed by a computer to store or process a data item.

There can be only one standard field defined for an element in each DDL. The physical format of the standard field must be identical to the format defined in the element edit string of the element. When any attributes of the standard field will not suffice, a nonstandard field must be defined.

The scope of this standard excludes any fields that are not implementations of a standardized element. Examples of fields that are excluded from this standard include processing-status flags (for example, "Marked for Deletion Indicator"), record-status values (for example, "Date of Last Backup") and so on. These data items, which do not contain "business" information, do not appear on data models of a business, and hence are not defined as elements, and so are not candidates for standardization as fields. Such fields do exist; and their definitions can

be expressed using the templates established in this standard; but their definitions cannot be standardized using this standard.

Field is a subtype of *Implementation Data Object*.

notes

A field is an addressable item of information within a computer data system. Fields and elements both define data items. Field definitions provide the information needed by computers to store or process the data item, while, element definitions provide the information needed by people to understand and control the data item.

The standard field definition of a business data item is always directly related to an element definition of that business data item. Many of the attributes in this standard definition are copied from corresponding attributes in the element definition. Refer to the definition of the *Defined as* relationship in Chapter 4 for specific information.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

related names

related name	context
CDD$DATA_ELEMENT	VAX CDD/Plus

IMPLEMENTATION DATA OBJECT

business name
Implementation Data Object

attributes

Data Object Identifier *[K]*
Data Object Revision *[K]*
Data Object Type
Data Object Status

description
The definition of some data object, that provides all the information that a computer-processing system requires about that data object to handle it within the implementation.

data object type
3NF Entity

abbreviated name
IMPLMN_DATA_OBJCT

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The requirements for creating implementation data objects are found as part of the definitions of entities Field, Standard Field, and Nonstandard Field.

Implementation Data Object is a subtype of *Data Object*.

notes
An implementation data object is any data object that has a role in the implementation of a system. At this time, there is only one subtype of implementation data object defined on the model, *Type Hierarchy of Data Object*; that subtype is *field*. Other types of implementation data objects, such as records, indexes, and files, may be added to this type hierarchy in a future revision.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

related names

related name	context
Standard Data Definition	Data Administration

NONSTANDARD FIELD

business name
Nonstandard Field

attributes

Data Object Identifier *[K]*
Data Object Revision *[K]*
Data Object Type
Data Object Status
Field Name
Field Label
Field Help
Field Datatype
Field Size
Field Query Name
Field Scale
Field Edit String
Field Justification
Input Validation Expression
Input Validation Routine
Computed-by Expression
Computed-by Routine
Missing Value
Initial Value

description
A locally modified version of a Standard Field.

data object type
3NF Entity

abbreviated name
NONSTD_FLD

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
A nonstandard field must be linked to a standard field through the *based on* relationship. A nonstandard field cannot be standardized until the standard field is standardized.

Through the *based on* relationship, the attributes of the standard field (except for Data Object Identifier, Data Object Revision, Data Object Type, and Data Object Status) are copied to the definition of the nonstandard field as its default values when it is created. Any of these copied attributes can be changed to suit the processing requirements of the local environment.

Nonstandard Field is a subtype of *Field*.

notes
A nonstandard field is a modified version of a standard field, created to meet local processing requirements. The data carried in a nonstandard field has the same business meaning as the data in a standard field;

any difference of business meaning must be conveyed by a new element definition.

The purpose of standardizing nonstandard fields is to provide the ability to control all fields that carry a particular piece of business data (that is, all fields that relate back to the same element). All fields that carry the same piece of business data must be related back to the same element. The one standard field definition that establishes the common definition must be used whenever possible. In addition, all other fields that carry this data must be defined as nonstandard fields for that element.

A nonstandard field is appropriate in many cases, where the business meaning of a data item is the same but the field characteristics must vary. For example the element *Employee Identifier* could have a standard field called *EMPLYE_ID*, which establishes the standard size and datatype of this piece of data. This element could be used throughout the optimized model, for example, wherever this item of business data is represented. But in an implementation, this data item might need to appear in many different files or contexts; and the implementation may require that it be given a different name in each. For instance it could appear in each employee record in a master employee file. In this instance a field called *MEF_EMPLYE_ID* could be defined as a nonstandard field. That name would distinguish this field from another field in, for example, a file of nightly change transactions where this data item might be called *MEF_TNSACT_EMPLYE_ID*. Note that this example is provided only as an illustration; such redefinition may not be required, and in fact it adds complexity and cost to the support of the implementation. But when such redefinition is used, it must be managed as a nonstandard field.

Another appropriate use of nonstandard field is to record field definitions in applications where the size, datatype, or other attribute of the field must differ from the standard field definition; a reason why this might be justified would be for a field that is used to communicate with pre-existing, noncompliant systems. Fields that are mapped in this manner must ultimately map back to standard elements that are defined in standard data models.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

related names

related name	context
CDD$DATA_ELEMENT	VAX CDD/Plus

NUMERIC FORMAT

business name
Numeric Format

attributes

Data Format Type *[K]*
Element Edit String
Maximum Character Count
Element Scale
Signed Numeric Indicator

description
The format of a data element that represents a numerical data item, usable in computations.

data object type
3NF Entity

abbreviated name
NUMRC_FRMT

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
A numeric format must be employed for any data element that is used in computation, including arithmetic expressions (for example, multiplication) and summary expressions (for example, averaging).

Numeric Format is a subtype of *Data Format*. It is a weak entity which depends entirely on the element to which it is connected for its identification.

notes
The numeric data format defines certain physical characteristics of an element which contains numeric data. The characteristics that are defined for a numeric format are those which the business needs to control or understand in order to use the data consistently. Other physical characteristics, which pertain only to the implementation of the element, are established through the field definition.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

OPTIMIZED ENTITY

business name
Optimized Entity

attributes

Data Object Identifier *[K]*
Data Object Revision *[K]*
Data Object Type
Data Object Status
Business Name
Description
Abbreviated Name
Integrity Narrative
Notes

description
A structured group of elements, including keys and attributes, that has been organized in such a way as to optimize the performance, reliability, ease-of-use, and/or security of a data base implementation. An optimized entity can include elements that result from a process analysis, as well as elements that appear on a corresponding 3NF data model.

data object type
3NF Entity

abbreviated name
OPTMZD_ENTY

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
An optimized entity always contains the same attributes, regardless of the model in which it appears. When the attribute list changes, a different optimized entity must be defined. The differences should be documented in the definition. In general, it is preferable to use (or revise) an existing definition rather than create a new one; not only is the resulting documentation less burdensome, but this will result in improved integration of implemented systems.

Optimized Entity is a subtype of *Entity*.

notes
An optimized entity is frequently a transformation from one of the 3NF entities or from complex relationships. An optimized entity may contain data that has been identified on a process model, in addition to the data that has been identified on a 3NF model. Also, an optimized entity can contain non-normalized features such as derived data, repeating groups, and foreign keys.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

OPTIMIZED MODEL

business name
Optimized Model

attributes

Data Object Identifier *[K]*	Business Name
Data Object Revision *[K]*	Description
Data Object Type	Abbreviated Name
Data Object Status	Integrity Narrative
	Notes

description
A data model that presents the same data content (or a subset) as that found on one or more 3NF models, with the data structures modified to optimize the performance, ease of use, or security of the data in a particular implementation. An optimized model may also reflect any additional data requirements that arise from the analysis of the processes or functions that correspond to this data and application domain.

data object type
3NF Entity

abbreviated name
OPTMZD_MODEL

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
In a standardized definition of an optimized model, the integrity narrative must identify the implementation objectives that have determined the optimizations made on this model. Every optimized model must be a transformation of one or more 3NF models; for each transformation, a transformation narrative must be supplied that identifies which pieces of the 3NF model have been carried forward.

Optimized Model is a subtype of *Data Model*.

notes
The optimized model is the data model that results from the transformation of a 3NF model into a data structure design that can be implemented without any further changes to the business meaning of any object.

The transformation process may involve denormalization of the entities and simplification of the relationships. These changes are based on factors other than the data itself, including such things as the data base management tools, the scope of interest of a particular application, the access patterns of the update and inquiry procedures, the processes and transformations that act on the data, the physical storage requirements, the processing speed of the hardware, and security.

Different data modeling and systems design techniques have different approaches to the process of transforming a 3NF data model into a data model that can be implemented, but all the techniques regard this optimization process as a necessary and critical step. The result of this transformation is generically known as an optimized model. However, the different techniques use different names. For example, in the Merise technique this model is known as a Functional Data Model. In the Information Engineering literature, this model is known as an Access Model.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

related names

related name	context
Functional Model	Merise
Access Model	Information Engineering
Application Model	Common Usage
Logical Model	Common Usage
Denormalized Model	Common Usage

RELATED NAME

business name
Related Name

attributes
Related Name *[K]*

description
The nonstandard aliases, synonyms, misnomers, or previously valid names for data objects.

abbreviated name
RELTD_NAME

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Related names may be added or removed from a data object's definition at any time, upon the approval of the object's controller.

notes
Related names are generally alternate, nonstandard names for a data object. These names may have originated in such contexts as third-party packages, external standards, existing applications, or common usage. Related names are sometimes the former standard name of the data object, when the former name continues to be used.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

related names

related name	context
Alias	Common Usage
Synonym	Common Usage
Misnomer	Common Usage
Keyword	Common Usage

RELATED NAME CONTEXT

business name
Related Name Context

attributes

Related Name Context *[K]*

description
The context in which a related name is used as a synonym for a data object.

data object type
3NF Entity

abbreviated name
RELTD_NAME_CNTXT

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Additional related name contexts can be created whenever a new related name is added. An existing context should be used whenever feasible in order to enhance the usefulness of this information.

notes
The related name context identifies a broad domain in which a set of related names are used. For example the related name context could be the name of an external standard, or the name of a third-party application.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

RELATIONSHIP

business name
Relationship

attributes

Data Object Identifier *[K]*
Data Object Revision *[K]*
Data Object Type
Data Object Status
Business Name
Description
Abbreviated Name
Reverse Relationship Name
Integrity Narrative
Notes

description
A defined interaction among entities that is of interest in the business area being modeled. A relationship defines the business rules that govern this interaction, including the nature and number of the participation by each entity. A relationship can join one entity with itself, in which case it is called *recursive*.

data object type
3NF Entity

abbreviated name
RLNSP

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Every Relationship definition must describe the business information that is represented by an occurrence of that relationship.

Relationship is a subtype of *Association*.

notes
Relationships in a data model denote a connection between entities. When a relationship is defined, the definition must specify the business or technical rules that control the association.

There are two types of relationships: simple relationships and complex relationships. A simple relationship does not have attributes, joins only two entities, and is not a many-to-many relationship. A complex relationship, on the other hand, is one which has attributes, or joins more than two entities, or involves a many-to-many relationship. Some analytic techniques state that 3NF Models must not contain any complex relationships; however, not all analytic techniques contain this

requirement. However, all analytic techniques agree that Optimized Models cannot contain complex relationships.

Different methodologies use different graphic symbols to represent relationships on data models. For example, the Merise technique uses an ellipse, the Yourdon technique uses a diamond, and the Bachman technique simply uses a connecting line. In all cases, the concept is the same, and the same set of information (such as a Business Name, Description, and Label) is collected about relationships regardless of technique.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

SECURITY CLASSIFICATION

business name
Security Classification

attributes

Security Classification Name *[K]*

description
The classifications of proprietary data or information, as defined in Digital Security policies.

data object type
3NF Entity

abbreviated name
SCURTY_CLSFTN

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The classifications of Proprietary Information are established by Corporate Information Security.

notes
Information security classifications are defined in Section 8.03, Proprietary Information Protection, of the *U.S. Personnel Policy and Procedures Manual*. This policy defines the classifications, requirements, and responsibilities for handling information in each class.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

SIMPLE RELATIONSHIP

business name
Simple Relationship

attributes

Data Object Identifier *[K]*
Data Object Revision *[K]*
Data Object Type
Data Object Status
Business Name
Description
Abbreviated Name
Reverse Relationship Name
Integrity Narrative
Notes

description
A relationship between entities that is binary, has no more than one maximum cardinality with a value greater than 1, and does not identify any element values.

data object type
3NF Entity

abbreviated name
SIMPL_RLNSP

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
A simple relationship must join two entities (or, recursively join an entity to itself). Any relationship that joins more than two entities is, by definition, a complex relationship.

A simple relationship must not involve a many-to-many relationship. In other words, no more than one of the maximum cardinalities can have a value greater than 1. If both maximum cardinalities are greater than 1, then the relationship between the two entities is complex.

A simple relationship cannot have attributes. If any elements are fully dependent on the concatenation of the keys of the associated entities, then the relationship is a complex relationship.

Simple Relationship is a subtype of *Relationship*.

notes
A simple relationship is a one-to-many relationship between two entities (or, if the relationship is recursive, it may join a single entity to itself). For example, in the model *Type Hierarchy of Data Objects* in Chapter 2 of this document, the relationship "Business Data Object is Controlled

by a Controller" is a simple relationship: it has a one-to-many maximum cardinality, it relates two entities, and it has no attributes of its own.

A simple relationship can usually be implemented as a pointer, a link, or a foreign key, rather than being implemented as an independently defined record (or Relation). Generally, any relationship that must be implemented as an independently defined record is a complex relationship.

Some analytic techniques require that all relationships on a 3NF model must be simple relationships. Frequently, additional clarity is achieved by simplifying all relationships at the 3NF level, but some methodologies permit complex relationships. DEC STD 065 allows either type of relationship to appear on a 3NF model.

On optimized models, it is not always necessary to explicitly define every access path between entities as a simple relationship. The access path may be left implicit if there is no unique, specific business information conveyed by the existence of the relationship. In such cases, a connecting line may appear on the diagram of the optimized model.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

STANDARD FIELD

business name
Standard Field

attributes

Data Object Identifier *[K]*
Data Object Revision *[K]*
Data Object Type
Data Object Status
Field Name
Field Label
Field Help
Field Datatype
Field Size
Field Query Name
Field Scale
Field Edit String
Field Justification
Input Validation Expression
Input Validation Routine
Computed-by Expression
Computed-by Routine
Missing Value
Initial Value

description
The field definition that serves as the standard for implementation of its element for a given DDL. The Standard Field definition is the definition that exactly matches the specifications of the element definition, subject to the limitations of the DDL.

data object type
3NF Entity

abbreviated name
STD_FLD

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
A standard field cannot be established without an element. An element represents an item of business data; therefore a standard field can only be standardized if it contains an item of business data. Elements in turn cannot be established without an entity, and entities cannot be established without a model. Hence the business data that the field contains must be modeled and standardized before the field itself can be standardized.

There can be only one standard field for an element in each Data Definition Language. New standard fields can be created only when the existing CDO Standard Field cannot be used by a chosen data description language.

The definition of the standard field of an element must match the data format specified by the element definition, subject only to the limitations of the DDL in which the field is defined. The values of each attribute of the standard field use, as their default, certain properties from the element definition. This correspondence is defined in Table 4–1, within the definition of the *defined as* relationship.

Standard Field is a subtype of *Field*.

notes

The standard field of an element is the field that should be used in all new implementations that involve a data element.

The standard field in the CDO language is the field that must be used for interchange of the element from one application to another. This format must match exactly the *element edit string* specified for the element.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

related names

related name	context
CDD$DATA_ELEMENT	VAX CDD/Plus

SUBTYPING

business name
Subtyping

attributes

Data Object Identifier *[K]*	Business Name
Data Object Revision *[K]*	Description
Data Object Type	Abbreviated Name
Data Object Status	Integrity Narrative
	Notes

description
A form of association that expresses an exhaustive, partitioned classification of an entity. Each occurrence of the entity must also be defined in terms of only one of its classifications.

data object type
3NF Entity

abbreviated name
SUBTYP

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The classification of an entity into subtypes must be exhaustive. This means that every occurrence of the supertype entity must be an occurrence of one of the subtypes.

The classification of an entity into subtypes must be partitioned. This means that every occurrence of the supertype must be an occurrence on only one of the subtypes.

A supertype can decompose into two or more distinct subtyping hierarchies. In this case, the rules in the preceding two paragraphs are applied within the bounds of each hierarchy.

A subtype entity inherits all the keys and attributes of the supertype, as well as all the relationships in which the supertype participates. The subtypes may also have additional attributes and participate in additional relationships that the supertype does not.

Subtyping cannot appear on an optimized model, because such an optimization would require technology that intrinsically supports inheritance. When the use of such technologies becomes standardizable, this restriction may be removed.

Subtyping is subtype of *Association*.

notes

The entity that is broken down into classifications is called the supertype entity; the classifications are called subtype entities.

The subtyping relationship provides a means for defining specific categories into which an entity is divided. These categories are defined as entities, and as such can participate in relationships in data models.

The subtyping relationship is also called an "IS-A" relationship, because every occurrence of the subtype is an occurrence of the supertype.

Subtyping may be transformed in a variety of ways when a model is optimized. For example, the supertype can be carried down to the optimized model intact, where it serves as the vehicle for the core attributes and relationships. Meanwhile the subtypes are carried down as optimized entities, discarding the nonkey attributes of the supertype; the subtypes are then each linked back to the supertype by a series of explicit, mutually exclusive simple relationships. Another optimization approach is to transform all the subtypes into a single optimized entity, that carries the complete set of attributes from all of the subtypes; in this example there would be no need for an entity corresponding to the supertype.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

related names

related name	context
IS-A	Object-Oriented Languages
Inheritance	Object-Oriented Languages

TEXT FORMAT

business name
Text Format

attributes

Data Format Type *[K]*
Element Edit String
Maximum Character Count
Variable Size Indicator
Text Justification

description
The format of a data element that contains textual, character-oriented data.

data object type
3NF Entity

abbreviated name
TXT_FRMT

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The text format defines certain physical characteristics of an element that contains character data. The characteristics that are defined in the data format are those that the business needs to control or understand in order to use the data consistently. Other physical characteristics, which pertain only to the implementation of the element, are established through the field definition. The text data format must be used for any element that has a variable length.

Text Format is a subtype of *Data Format*. It is a weak entity that depends entirely on the element to which it is connected for its identification.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

4 STANDARD ASSOCIATION DEFINITIONS

This chapter contains the definitions of each of the associations that appear on the two models in Chapter 2, *Standard Data Model Requirements*, and *Type Hierarchy of Data Objects*.

Each association between entities represents some connection or correlation that pertains to the complete definition of a data object.

AGGREGATE OF

business name
Aggregate Of

entities associated and cardinalities

Data Object	0,N
Data Object	0,N

attributes

Aggregation Narrative

description
The association between two data objects wherein a component object refers to some portion of the overall structure in a group object.

data object type
Complex Relationship

abbreviated name
AGRGT_OF

reverse relationship name
Component Of

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The objects related by this relationship must be the same data object type, except that an optimized entity can consist of 3NF entities.

The controller of the parent data object has the authority to approve the creation of any defined portion of the data object.

notes
The *aggregate of* relationship links a group data object with its structural components. A component data object, in this relationship, is a defined subset of the structure of the group object, not a subset of its occurrences.

The *aggregate of* relationship can be explained in terms of each type of data object. If an element is an aggregate of other elements, then it is a *Group* or *Aggregate* element and the subelements refer to a set of the characters within the aggregate. The aggregation narrative can explain the relative order of the components, or any special formatting requirements.

For fields, the relationship means the same as it does for elements: it is a breakdown of an aggregate into its constituent subfields.

Any element or field that has no components is called an *Atomic* element or field.

When an entity is an aggregate of other entities, then the subentities are usually called *Segments*. Aggregation in this context means that the attribute list of the composite entity is made up of the attribute lists of some or all of its segments. The aggregation narrative can describe the business rules that govern the participation of a segment within a composite entity.

For associations, the *aggregate of* relationship is undefined.

For models, the subpieces of a model are often referred to as *Facets* or *Subjects*, depending on the methodology being employed. Each portion represents a subset of the entities and associations that make up the overall model.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

related names

related name	context
Segment	Transaction Standards
Group Element	Common Usage
Aggregate	DECdesign

ASSOCIATES

business name
Associates

entities associated and cardinalities

Association	1,N
Entity	1,N

attributes

Maximum Cardinality

Minimum Cardinality

description
The linkage between an association and the entities that are associated through it

data object type
Complex Relationship

abbreviated name
ASSCT

reverse relationship name
Associated Via

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The rules for this relationship differ depending on the type of association involved.

A *simple relationship* can associate only two entities, whereas other types of association can link more than two entities.

Complex relationships, Classification and *Subtyping* can associate two or more entities. Because a complex relationship can appear only on a 3NF model, only 3NF entities may be associated in a complex relationship. The same is true for *subtyping*; all entities involved in a subtyping relationship must be *3NF entities*.

notes
An association is not uniquely defined without reference to the entities it associates.

The *associates* relationship denotes the connection between the relationship and the entities associated. In the Merise graphic conventions, the *associates* relationship refers to the connector between the relationship (an ellipse) and the entity (a roundtangle). There is one *associates* relationship for each entity that participates in the relationship.

The cardinalities of an entity within a relationship is specified by the *associates* relationship.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

related names

related name	context
Connector	DECdesign

BASED ON

business name
Based On

entities associated and cardinalities

Nonstandard Field	1,1
Standard Field	0,N

description
A link that allows a nonstandard field definition to share the attributes of a standard field that is already defined.

data object type
Simple Relationship

abbreviated name
BASED_ON

reverse relationship name
Has Local Variant

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The *based on* relationship may only be established to support the definition of a nonstandard field. When a nonstandard field is created, it must be based on a standard field. The attributes of the standard field become the initial values of the attributes of the nonstandard field. Any of these attributes can then be changed.

A nonstandard field cannot be based on another nonstandard field.

notes
When a nonstandard field is based on a standard field, the nonstandard field copies the values of certain attributes of the standard field. This provides a starting point for the definition of the nonstandard field; the values of any of the attributes can be changed. The following attributes are copied to the new field definition:

Field Name
Field Help
Field Label
Field Datatype
Field Size
Field Query Name

Field Scale
Field Edit String
Field Justification
Input Validation Expression
Input Validation Routine
Computed-by Expression
Computed-by Routine
Missing Value
Initial Value

The present definition of this relationship in the context of DEC STD 065-0 does not express the complete usage or potential of the *based on* relationship as it is implemented in CDD/Plus. The present definition is not intended to limit any repository system from taking advantage of this potential. Instead these limitations in the definition reflect the current scope of the DEC STD 065-0 data models, as described in Chapter 2 of this standard. In particular, the present definition does not allow the definition of one field to obtain its attributes purely by reference to the other field.

The *based on* relationship is not to be used to mirror a *subclass of* relationship that exists between two elements. The standard fields that are defined for a parent element (for example, "Part Number") and its subclass element (for example, "Component Part Number") are each defined with independent *defined as* relationships (see the definition of *defined as* in this chapter); they do not have a *based on* relationship between them. The consistency of field definition between two such fields is guaranteed because both fields take their derived values from the same *Data Form*, which in this example would be the data form that belongs to the parent element, "Part Number."

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

related names

related name	context
BASED ON	VAX CDD/Plus

CLASSIFIED BY

business name
Classified By

entities associated and cardinalities

Business Data Object	0,N
Business Term	0,N

description
The classification of business data objects using defined business terms

data object type
Complex Relationship

abbreviated name
CLSSF_BY

reverse relationship name
Identifies a Class of

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
A business data object can only be classified using formally defined business terms. The process for creating business term definitions is detailed in the DDNG.

notes
The *classified by* relationship allows for the grouping of any selection of business data objects under a defined business term. The business terms used in this context do not need to appear in the business names of these or any other data objects; however, the terms must meet the requirements of the Business Term Glossary as defined in the DDNG.

Some analytic methods use the concept of "Subject Area" as a tool for identifying the general categories of data within a business. Subject area is not explicitly defined as a data object type in the current revision of DEC STD 065-0; however, the *classified by* relationship can be used to record the subject area of a business data object, if the subject area is first established as a defined business term.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

CONSISTS OF (Code Set)

business name
Consists Of

entities associated and cardinalities

Code Set	0,N
Code Value	1,N

description
The list of code values that make up a defined code set.

data object type
Complex Relationship

abbreviated name
CNSIST_OF

reverse relationship name
Belongs to

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The addition or removal of values in a code set is done under the authority of the controller of the element for which the code set is used. A code set can only be shared among elements that are each subclasses of a common element. The controller of the common element becomes the controller of the shared code set.

A code value can appear on more than one code set, but only if the elements involved share a common ancestor through the *subclass of* relationship. This requirement guarantees that the elements share a common data form, which in turn is the data form of the code value items.

A code set can refer to other code sets through the *subset of* relationship (see its definition in this chapter). Code values should be identified in one way or the other; that is, if a code set calls out another code set as a subset, then the code values of the subset should not be repeated as code values of the superset.

If a code set refers to subsets only, then it will not point to any code values. Note however that a code set can contain a mixture of other code sets plus individual code values.

notes

This relationship establishes the individual entries in a list of valid code values.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

CONSISTS OF (Data Model)

business name
Consists Of

entities associated and cardinalities

Data Model	1,N
Association	1,N
Entity	1,N

description
The identification of the entities and associations that appear in a data model.

data object type
Complex Relationship

abbreviated name
CNSIST_OF

reverse relationship name
Appears on

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Every association must appear on at least one model, otherwise it cannot be standardized. An optimized model does not have to contain associations, depending on what the objectives of the optimization and the modeling technique require.

The types of association that may appear on a 3NF model depend in part on the modeling technique being used. For example, some techniques do not permit the use of subtyping on 3NF (or any) models. Similarly, some techniques do not permit complex relationships on 3NF models.

The only type of association that may appear on an optimized model is a simple relationship.

notes
A data model consists of entities and associations, regardless of whether it is a 3NF model or an optimized model. A 3NF model can consist of any sort of associations, and can only contain 3NF entities. An optimized model, on the other hand, can contain any sort of entity and can only contain simple relationships.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

CONTROLLED BY

business name
Controlled By

entities associated and cardinalities

Business Data Object	1,1
Controller	0,N

description
The identification of the agent or agency that is responsible for the accuracy and usability of the definition of a business data object.

data object type
Complex Relationship

abbreviated name
CTRL_BY

reverse relationship name
Responsible for

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The responsibilities and authority of the controller are defined in DEC STD 065-0 Section 1.4.2.

This relationship represents the fact that a legitimate controller exists for the associated business data object, and this controller endorses the content of the definition.

Once a controller of a data object has been established, the controller has complete authority to determine the content of the definition. However, the determination of a proper controller is subject to review as part of the process of determining whether a data object definition meets the requirements of this standard.

The controller of an approved definition can be changed through the mutual consent of the outgoing and incoming controllers. Such a change is subject to review by the EDB to ensure that the new controller meets the qualifications defined in this standard. Such a change does not require a revision to the data object; however the new controller may elect to accept responsibility only after the object definition has been revised.

notes

This relationship identifies the controller of a business data object. Every business data object must have a controller. Implementation data objects do not have their own controllers; they are the responsibility of the controller of the business data object they implement.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

CREATED FOR

business name
Created For

entities associated and cardinalities

Field	1,1
Data Description Language	0,N

description
The identification of the DDL for which a field definition is written.

data object type
Simple Relationship

abbreviated name
CREATD_FOR

reverse relationship name
Is the Environment for

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Every field must be created for a single DDL.

In a single DDL, the first field that is created for an element must be the *standard field*. This field must meet the requirements for standard fields as described under the *defined as* relationship in this chapter. Subsequently, every other field that is defined for the same element in the same DDL must be a *nonstandard field*.

notes
The principal language for defining fields in Digital is CDO, the Common Dictionary Operator utility within VAX CDD/Plus. Fields created for CDO can be used by any language or data manipulation utility that is supported by CDD/Plus.

However, there may be some language-specific requirements that prevent the CDO field from being used in that language. For example, the Pascal language prohibits underscores within field names so default field names will not work in Pascal applications. When the language-specific requirements dictate that the CDO field cannot be used, then a new standard field must be created for that additional language environment.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

DEFINED AS

business name
Defined As

entities associated and cardinalities

Element	0,N
Standard Field	1,1

description
The relationship between an element and a standard field that has been defined for that element.

data object type
Simple Relationship

abbreviated name
DFN_AS

reverse relationship name
Implements

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Every *standard field* must be related to one and only one element definition.

For a single DDL, there can be only one *standard field* for each element. A single element can have more than one *standard field*, but they must all be in different DDLs and they must all conform to the requirements outlined in Table 4–1.

The definition of a standard field must correspond to the definition of the element it implements. Table 4–1 shows these correspondences. The first portion of the table shows the pairs of properties that must have a direct correspondence. For these pairs, the field's attribute must be identical to the element's attribute *subject to the limitations of the host DDL*. For example, the *field label* must match the element's *label* unless the DDL cannot accept the element's *label* (for example, the length of the terms in the element's label might exceed the limits of the DDL).

The second portion of Table 4–1 shows the pairs of properties that have an indirect correspondence. With these properties, the field's property must be consistent with the the element's property but the actual value of the element's property is not suitable for an implementation data object. For example, the field's *input validation* must be a DDL compatible implementation of the rules described in the element's *integrity narrative*.

Table 4–1 Correspondence of Standard Field to Element

Element Attribute	Field Attribute
Direct Correspondence	
Abbreviated Name	Field Name
Business Name	Field Query Name
Size in Characters	Field Size
Element Edit String	Field Edit String
Element Scale	Field Scale
Text Justification	Field Justification
Indirect Correspondence	
Business Name	Field Label
Description	Field Help
Data Form Type	Field Datatype
Integrity Narrative	Input Validation Expression
Integrity Narrative	Input Validation Routine
Derivation Narrative	Computed By Expression
Derivation Narrative	Computed By Routine
Default Action Narrative	Initial Value
Default Action Narrative	Missing Value

Refer also to the definition of the *Based on* relationship, in this chapter. The *Based on* relationship defines how the properties of a standard field are the basis for the definition of a nonstandard field, whereas the correspondences in Table 4–1 define how the properties of an element are the basis for the definition of a standard field.

The controller of an element must approve of the standard field definition that is defined for that element.

Every element on an optimized model must have at least one field defined for it in the CDO DDL.

notes

The *defined as* relationship identifies the standard field definitions for an element. In a single DDL, there can be only one *standard field* for each element.

The business meaning of a field cannot differ from that of the element it implements. Variations of business meaning cannot, for example, be built into the name of a field.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

HAS ATTRIBUTE (Complex Relationship)

business name
Has Attribute

entities associated and cardinalities

Complex Relationship	0,N
Element	0,1

description
The identification of any element that is an attribute of a complex relationship.

data object type
Simple Relationship

abbreviated name
HAS_ATTRBT

reverse relationship name
Is an Attribute of

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
An element can appear as an attribute on only one complex relationship.

An element that is an attribute of a complex relationship can also appear as an attribute of a 3NF entity on a different model, but only if the primary key of the 3NF entity is the concatenation of the primary keys of all the entities that are associated through the complex relationship. This exception allows different methodologies to capture the same element, with the same unique primary key.

notes
This relationship describes the construct whereby complex relationships can have attributes on 3NF models. This construct is available in the Merise method, and is usable on any standardized model subject to the requirements of the technique used to normalize the model.

Any relationship that has attributes is by definition a complex relationship and cannot appear on an optimized model.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

related names

related name	context
Property	Data Administration

HAS ATTRIBUTE (Entity)

business name
Has Attribute

entities associated and cardinalities

Entity	0,N
Element	0,N

description
The link between an entity and an element that specifies the set of elements that apply to that entity.

data object type
Complex Relationship

abbreviated name
HAS_ATTRBT

reverse relationship name
Is an Attribute of

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Each element must be associated with at least one entity as either an attribute or key, or with a complex relationship as an attribute. An element definition cannot be standardized outside the context of one of these modeling relationships.

An element cannot be listed as both the key and an attribute of the same entity.

An element cannot be an attribute of more than one 3NF entity. An element that is an attribute of a 3NF entity cannot be part of the key of any 3NF entity.

The elements that are attributes of a 3NF entity must conform to the rules of third normal form. These requirements are stated in the documentation of any data modeling technique, and can be summarized as follows:

1 All elements must be organized into entities that have a primary key; the value of the element must apply to the object referred to by that primary key.

2 No element can appear more than once in any entity; there cannot be any "repeating groups" of elements.

3 Every element must depend on the complete primary key; there cannot be dependence on just a portion of the key.

4 Every element must depend entirely on the key and on no other attributes of the entity; there cannot be any transitive dependence among attributes.

The rules for normalization of data focus specifically on the required relationships between an element and the 3NF entity it appears on. When a model is optimized, and optimized entities are created, these rule are relaxed. Note, however, that denormalization of elements always diminishes the integrity and/or flexibility of a data structure. These costs must always be weighed against the benefits that are sought by the optimization, such as performance, security, and reliability.

notes

The term "attribute" is frequently used to refer to the elements that are attributes of an entity. There is no difference in the meaning of "attribute" and "element" other than that an attribute is an element that is associated with some entity. The term "property" is occasionally used as a synonym for "attribute" in this sense, particularly when the element is a piece of metadata. Thus the attributes of Entities, Elements, Associations, and Data Models are sometimes referred to as "properties" of data objects.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

related names

related name	context
Attribute	Data Administration
Property	Data Administration

HAS DATA FORM

business name
Has Data Form

entities associated and cardinalities

Element	0,1
Data Form	1,1

description
The linkage between an element and the specification of its data form.

data object type
Simple Relationship

abbreviated name
HAS_DATA_FORM

reverse relationship name
Is the Format for

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Every element must participate in either the *has data form* relationship or the *subclass of* relationship. The *subclass of* relationship allows a single data form to be shared by a family of elements. When a subclass is used, it must eventually resolve to an element that has a Data Form specified; such an element is sometimes called the root element.

An element can have only one defined data form. This defines the standard data form of that element, independent of language or application.

notes
The *has data form* relationship defines the standard physical characteristics of an element, such as the physical format in which the element is shared across application boundaries.

When an element is defined as a subclass of another element, it adopts the data form of the superclass element. This is similar to the concept of *based on* in field definitions, except that among elements there is a requirement for a logical as well as a physical congruence.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

related names

related name	context
Based On	DEC STD 065 Rev C

HAS DATA SECURITY

business name
Has Data Security

entities associated and cardinalities

Business Data Object	1,1
Security Classification	0,N

description
The identification of the corporate security classification that applies to a business data object. This is the security of the data object's data value.

data object type
Simple Relationship

abbreviated name
HAS_DATA_SCURTY

reverse relationship name
Defines Data Security of

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
By default, all data definitions have the classification, Digital Internal Use Only. The rules for assigning data to a classification are defined in *U.S. Personnel Policy Manual* number 8.03 Proprietary Information Protection, and in other Security, Engineering, and Digital Information System (DIS) policies. These policies define the classifications and how to assign data to them, and lay out the requirements and responsibilities for handling information in each class.

notes
This relationship associates a data definition with the security classification that applies to the values of the data defined.

The security classification of data does not directly determine the classification of the data definition. For example, a data definition taken from the public domain, such as an element in an EDIFACT transaction, would be classified as nonproprietary. However, the data values could be Digital's proprietary information; for example, if the element identified the dollar value of a particular transaction.

The security classification identified through this relationship specifies the general proprietary level of the business information. The classification does not specify the level of security appropriate to any specific implementation data object, such as a record within a data base.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

HAS METADATA SECURITY

business name
Has Metadata Security

entities associated and cardinalities

Business Data Object	1,1
Security Classification	0,N

description
The identification of the corporate security classification that applies to a piece of metadata; that is, to the business data object definition as a business document.

data object type
Simple Relationship

abbreviated name
HAS_MTDATA_SCURTY

reverse relationship name
Defines Metadata Security of

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
By default, all data definitions have the classification, Digital Internal Use Only. The rules for assigning a definition to a classification are defined in *U.S. Personnel Policy Manual* number 8.03 Proprietary Information Protection, and in other Security, Engineering, and DIS policies. These policies define the classifications and how to assign data to them, and establish the requirements and responsibilities for handling information in each class.

notes
This relationship associates a data definition with the security classification that applies to the definition itself.

The classification of a data definition does not directly determine the classification of the data defined. For example, a data definition taken from the public domain, such as an element in an EDIFACT transaction, would be classified as nonproprietary. However, the data values could be Digital's proprietary information; for example, if the element identified the dollar value of a particular transaction.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

IS AN ASSOCIATION SUBTYPE

business name
Is An Association Subtype

entities associated and cardinalities

Association	1,1
Classification	1,1
Subtyping	1,1
Relationship	1,1

description
The differentiation of Associations according to how the entities correspond to one another.

data object type
Subtyping

abbreviated name
ASSCTN_SUBTYP

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Within the boundaries of DEC STD 065-0, every association must be a Classification, a Subtyping, or a Relationship. A Classification is a specialization of an entity into categories that have additional attributes or additional business rules that apply to them. A Subtype is a specialization of an entity into categories that are exclusive and exhaustive, and which also have differences in form. A relationship is an interaction between two or more entities that are of interest to the business.

Two of the subtypes of Association correspond directly to values of the element Data Object Type: Classification and Subtyping. Relationships are identified as having one of the following values of Data Object Type: Simple Relationship or Complex Relationship.

notes
Additional types of association may exist, but are beyond the consideration of this revision of the definition.

When using the subtypes of Association, it is important to note that Relationships differ significantly from the two other types. These two, Classification and Subtyping, express specialization of an entity into a further, more detailed, level of interest. An individual occurrence of the child entity in either of these types of association is simultaneously an occurrence of its parent type. Borrowing the example from Chapter 2 of this standard, the occurrence of an "Employee" entity is simultaneously an occurrence of a "Worker" entity. There is only one object (the individual person), and it is an occurrence of both entities. Because of the parent-child hierarchy obtained in such an association, any attributes of the parent entity are necessarily also attributes of the child entity; this characteristic is called inheritance, and it applies in Subtyping and Classification.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

IS A BUSINESS DATA OBJECT SUBTYPE

business name
Is A Business Data Object Subtype

entities associated and cardinalities

Business Data Object	1,1
Data Model	1,1
Entity	1,1
Association	1,1
Element	1,1

description
The differentiation of Business Data Objects according to their granulation.

data object type
Subtyping

abbreviated name
BUSNS_DATA_OBJCT_SUBTYP

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Within the boundaries of DEC STD 065-0, every business data object must be either a Data Model, an Entity, an Association, or an Element. An element is an individual unit of meaningful data; an entity is an object of interest to the business, to which a specific list of elements apply; an association is a manner in which entities can be related, which again is of interest to the business; and a data model is a formal description of the data within a complete, bounded business domain.

The categorization of a business data object into these subtypes is determined by value of the element, Data Object Type. The following table shows the subtype into which each value of Data Object Type falls, for all business data objects:

subtype	data object type
Data Model	3NF Model Optimized Model
Entity	3NF Entity Optimized Entity
Association	Subtyping Classification Simple Relationship Complex Relationship
Element	Element

notes

Additional types of business data object exist, but are beyond the consideration of this revision of the definition.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

IS A DATA FORMAT SUBTYPE

business name
Is A Data Format Subtype

entities associated and cardinalities

Data Format	1,1
Text Format	1,1
Numeric Format	1,1
Date Format	1,1

description
The differentiation of Data Formats according to the usage of the data.

data object type
Subtyping

abbreviated name
DATA_FRMT_SUBTYP

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The standardizable data formats are Text Format, Numeric Format, and Date Format. The Date Format provides the information needed when establishing the format of an element that contains a date, a relative or absolute time, or a combination of date and time. The Numeric Format provides the information needed when establishing the format of an element that contains numeric data that is to be used in computations. The Text Format provides the information needed when establishing the format of an element that contains character-oriented data.

The particular subtype of a data format is indicated by the value of the element, Data Format Type. The value of this element corresponds to the name of each subtype: *Text, Numeric,* and *Date*.

notes
Additional data formats exist, but are beyond the consideration of this revision of the definition. Extensions to include other nonoverlapping formats, such as images or voice data formats, may be added to this subtype hierarchy in the future.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

IS A DATA MODEL SUBTYPE

business name
Is A Data Model Subtype

entities associated and cardinalities

Data Model	1,1
3NF Model	1,1
Optimized Model	1,1

description
The differentiation of Data Models according to the goals for which they are developed, and accordingly the rules against which they are validated.

data object type
Subtyping

abbreviated name
DATA_MODEL_SUBTYP

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Within the boundaries of DEC STD 065-0, every data model must be either a 3NF Model or an Optimized Model. A 3NF model is a model in that all of the data has been normalized into Third Normal Form (see Section 1.6.2). An Optimized Model is a model which transforms the data that was modeled on one or more 3NF models into a form that optimizes the organization of data based on the performance requirements, access patterns, ease of use, security, or software environment of an particular implementation target.

The names of the subtypes of Data Model correspond directly to values of the element Data Object Type, for the two types of Data Models: 3NF Model and Optimized Model.

notes
Additional types of data model exist, but are beyond the consideration of this revision of the definition.

The terms 3NF Model and Optimized Model are used in DEC STD 065-0, rather than other possible terms, to avoid misinterpretation of potentially ambiguous terms. Certain modeling terms, such as "Conceptual Model", are used differently in various modeling methodologies (in some methodologies, a conceptual model must be normalized; in others, a conceptual model is a prenormalized analysis of the business into

general data domains). In a similar vein, different methodologies use different terms to refer to the optimized model, even though most methodologies share in the idea that a 3NF model is usually altered prior to implementation to accommodate performance considerations and similar implementation issues. These types, 3NF Model and Optimized Model, are not new categories beyond those that are defined in popular methodologies; they are intended as independent terms for identifying the type of model, independent of the methodology used to develop it.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

IS A DATA OBJECT SUBTYPE

business name
Is A Data Object Subtype

entities associated and cardinalities

Data Object	1,1
Business Data Object	1,1
Implementation Data Object	1,1

description
The differentiation of Data Objects according to the ground of their meaning.

data object type
Subtyping

abbreviated name
DATA_OBJCT_SUBTYP

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Within the boundaries of DEC STD 065-0, every data object must be either a Business Data Object or an Implementation Data Object. A business data object is an item of information that is of interest to the business; its definition consists of all the information that a person needs to know about that data object to be used within their business. An implementation data object is an item of information that is of interest when implementing a computerized information system; its definition consists of all the information that the computer-processing system will use to handle that item.

The categorization of a data object into these subtypes is determined by value of the element, Data Object Type. The following table shows the subtype into which each value of Data Object Type falls, for all data objects:

subtype	data object type
Business Data Object	3NF Model Optimized Model 3NF Entity Optimized Entity Subtyping Classification Simple Relationship Complex Relationship Element
Implementation Data Object	Standard Field Nonstandard Field

The following values of Data Object Type indicate that the data object is an Implementation Data Object:

notes

Additional types of data objects exist, but are beyond the consideration of this revision of the definition.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

IS AN ENTITY SUBTYPE

business name
Is An Entity Subtype

entities associated and cardinalities

Entity	1,1
3NF Entity	1,1
Optimized Entity	1,1

description
The differentiation of Entities according to analytical rules against which they can be validated.

data object type
Subtyping

abbreviated name
ENTY_SUBTYP

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Within the boundaries of DEC STD 065-0, every entity must be either a 3NF Entity or an Optimized Entity. A 3NF entity is an entity in which all of the data has been normalized into Third Normal Form. An Optimized entity is an entity into which the data has been organized to optimize the performance, reliability, ease-of-use, security, or similar dimension of a particular implementation target.

The names of the subtypes of Entity correspond directly to values of the element Data Object Type, for the two types of Entities: 3NF Entity and Optimized Entity.

notes
Additional types of entities might exist, but they are beyond the consideration of this revision of the definition.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

IS A FIELD SUBTYPE

business name
Is A Field Subtype

entities associated and cardinalities

Field	1,1
Standard Field	1,1
Nonstandard Field	1,1

description
The differentiation of Fields according to their sharability.

data object type
Subtyping

abbreviated name
FLD_SUBTYP

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Within the boundaries of DEC STD 065-0, every field must be either a Standard Field or a Nonstandard Field. A Standard Field is the direct translation of a business data element into a particular DDL; for each element, there is only one standard field in a given DDL. A Nonstandard Field is a locally modified version of a standard field, created to meet local processing requirements. A standard field can be shared throughout the corporation, but a nonstandard field can only be shared when local processing requirements prevent the use of the standard field.

The names of the subtypes of Field correspond directly to values of the element Data Object Type, for the two types of Fields: Standard Field and Nonstandard Field.

notes
Additional types of fields exist, but are beyond the consideration of this revision of the definition.

Nonstandard Fields are included in DEC STD 065-0 because they contain business data, and therefore must be linked in a standard way with the element definitions of that business data. These definitions are standardizable so that they can be cataloged and reused among applications that share similar local processing requirements. The ability to share these definitions is not intended to imply that the data should

be shared in this format; the interchange of data must adhere to the Standard Field definition.

DEC STD 065-0 requires the standardization of all fields that contain business data, but does not permit the standardization of any fields that do not contain business data (in other words, fields that are not related to elements). The definitions of such fields could utilize the templates and tools associated with the standardizable fields from DEC STD 065-0; they would lack only two things: the validation gained by their appearance on an approved model and a connection to a responsible controller.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

IS AN IMPLEMENTATION DATA OBJECT SUBTYPE

business name
Is An Implementation Data Object Subtype

entities associated and cardinalities

Implementation Data Object	1,1
Field	1,1

description
The differentiation of Implementation Data Objects according to their granulation.

data object type
Subtyping

abbreviated name
IMPLMN_DATA_OBJCT_SUBTYP

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Within the boundaries of DEC STD 065-0, every implementation data object must be a field. A field is an identifiable unit of addressable data for computer processing.

The present boundaries of the data models defined by DEC STD 065-0 include only one type of implementation data object: fields. The following values of Data Object Type represent fields:

Standard Field

Nonstandard Field

notes
Additional types of implementation data object exist, but are beyond the consideration of this revision of the definition. The subtyping of Implementation Data Object into Fields is included in DEC STD 065-0 only for the purpose of providing a foundation for further expansion of this subtyping to include other types of implementation data objects.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

IS A RELATIONSHIP SUBTYPE

business name
Is A Relationship Subtype

entities associated and cardinalities

Relationship	1,1
Simple Relationship	1,1
Complex Relationship	1,1

description
The differentiation of Relationships according to the complexity of the connection they represent.

data object type
Subtyping

abbreviated name
RLNSP_SUBTYP

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Within the boundaries of DEC STD 065-0, every relationship must be a Simple Relationship or a Complex Relationship. A Complex Relationship is one that has one or more of the following characteristics:

1 More than two entities are related through it

2 It represents a many-to-many cardinality or

3 It has dependent attributes

A Simple Relationship is one that has none of the above characteristics.

The subtypes of Relationship correspond directly to the values of the element Data Object Type that apply to relationships: Simple Relationship and Complex Relationship.

notes
Additional types of relationships may exist, but are beyond the consideration of this revision of the definition.

Certain data modeling methodologies may prohibit some forms of complex relationship from appearing on 3NF Models. The rules of the particular methodology being employed should supersede the rules of this standard, insofar as the determination of analytic validity for that methodology is concerned.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

KNOWN AS

business name
Known As

entities associated and cardinalities

Business Data Object	0,N
Related Name	1,N
Related Name Context	0,N

description
The identification of the synonyms, alias, and misnomers by which a data object is known in various contexts.

data object type
Complex Relationship

abbreviated name
KNWN_AS

reverse relationship name
Is an Alias for

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
New entries may be made in an object's list of related names upon approval by the object's controller. Such changes do not result in new versions of the object's definition.

notes
A list of related names is a helpful mechanism to allow cross-references into a repository. These references can occur using a specific context or a specific related name.

If a controlled system of classifications is appropriate, the *classified by* relationship to specific, defined business terms should be employed rather than the *known as* relationship.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

NAMED USING

business name
Named Using

entities associated and cardinalities

Business Data Object	1,N
Business Term	0,N

description
The association between a business data object and the defined business terms that are used to form the object's business name.

data object type
Complex Relationship

abbreviated name
NAMED_USNG

reverse relationship name
Appears in the Name of

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The DDNG defines the complete set of procedures and requirements for creating business names of business data objects. One of the requirements is that every business term in the name must be defined in the Business Term Glossary and be assigned a standard abbreviation. The *named using* relationship identifies the business terms that appear in a business data object's name.

Only the business terms that appear in the business name shall be identified through this relationship. This includes those business terms that appear in the business name but do not appear in the abbreviated name. Any other business terms that apply to the data object are identified through the *classified by* relationship.

notes
Refer to the DDNG for a complete discussion of the requirements, procedures, and guidelines for naming business data objects.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

PARTICIPATES IN PRIMARY KEY OF

business name
Participates In Primary Key Of

entities associated and cardinalities

Element	0,N
Entity	1,N

description
The identification of the element or elements that comprise the primary key of an entity.

data object type
Complex Relationship

abbreviated name
PRTCPTS_IN_PRMRY_KEY_OF

reverse relationship name
Is Identified Using

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The rules for this relationship differ somewhat depending on the type of entity involved and the type of model the key appears on. These rules are as follows.

An element that participates in the primary key of a 3NF entity cannot appear as an attribute of any other 3NF entity. However, a single element can appear as a part of the key of more than one 3NF entity.

No two 3NF entities can have the same primary key. This requirement spans all standardized 3NF entities, not just entities on the same model. On an single optimized model, no two optimized entities can have the same primary key.

Every element must be linked with an entity, through one of the relationships *participates in primary key* or *has attribute*. No element definition can be standardized outside the context of an entity, which in turn must be standardized in the context of a data model.

notes
The primary key is the element or combination of elements that uniquely identify each occurrence of the entity.

Some entities, known as weak entities, depend on the identity of another entity for the uniqueness of their key. For example, in a human resource model the entity *Dependent* can be a weak entity that depends on the identity of *Employee*. Thus the dependent's name can be the primary key. Although the name might not be unique by itself, this name uniquely identifies the dependent once the employee's identity has been established.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

REPLACES

business name
Replaces

entities associated and cardinalities

Data Object	0,N
Data Object	0,N

attributes

Replacement Narrative

description
The association through which one data object replaces another. Both the replacement of an old revision by a new revision and replacement of any existing data object by a different data object are identified through this relationship.

data object type
Complex Relationship

abbreviated name
REPLCS

reverse relationship name
Has Been Replaced By

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The *data object status* of a definition that is replaced must become "Obsolete." This means that existing models that refer to an obsolete definition must begin to migrate to the replacement definition, and that new references cannot be created to the obsolete definition.

A single definition can be simultaneously replaced by two or more definitions. The circumstances for simultaneous replacement must be explained in the replacement narrative of each replacement relationship.

notes
The replacement of one data definition by another makes the original definition obsolete. An obsolete definition cannot be added to any new modeling structures. The obsolete definition remains available for existing references, until all existing references have migrated to the replacement.

The definition that replaces another is not necessarily capable of supporting all existing references to the original definitions. All existing references must be individually evaluated and judged before they are upgraded to the new reference. In some cases, the older definition must be replaced by two or more definitions in order to adequately support the existing references and the migration to the new definition. The replacement narrative shall indicate the guidelines for adopting each replacement object.

Every new *revision* of a data object definition requires that the new revision replaces the old revision. The concept of revision is a subset of the notion of replacement, where an existing data object definition is being changed but the object it refers to is essentially the same. When the new revision is approved, the status of the old revision becomes *Obsolete*.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

SUBCLASS OF

business name
Subclass Of

entities associated and cardinalities

Element	0,1
Element	0,N

description
The relationship between a data element that refers to some selected subset of data values, and the data element that refers to the full set of data values from which the subset is identified.

data object type
Simple Relationship

abbreviated name
SUBCLAS_OF

reverse relationship name
Has Subset

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
An element must participate in either the *has data form* relationship or the *subclass of* relationship, but not both. The child element adopts the data form of the parent element. If the parent is also a child of some other element, then the chain of *subclass of* relationships must eventually terminate in an element that has a data form.

The need for use of a *subclass of* relationship will only arise when a single element appears in multiple modeling contexts and the rules that govern its behavior in those contexts must be distinguished. A subclass element cannot be justified without referring to the modeling context in which it is being defined or used.

Wherever possible, the root element should be used. The *subclass of* relationship must not be used purely to reflect the name of function, organization, geography, or application in which the element is used, or purely to reflect its participation in more than one host entity. The *subclass of* relationship cannot be used to establish formal equivalence (that is, sharing of data form) unless the occurrences of the child element are by definition occurrences of the parent.

A child element cannot be created to represent an element that is inherited through Subtyping or Classification. For example, the entity Trading Partner could be defined with the key Trading Partner Identifier. If the entity Customer is then defined as a category of Trading Partner, its key is Trading Partner Identifier (and not some new subclass such as Customer Identifier).

The naming procedure defined by the DDNG specifically prevents the creation of subclasses based on the name of the organization that creates or uses the data, the name of the geography where the data is created or used, or the name of the application where the data is created or used. An element can be a valid subclass only if it conforms to these prohibitions. Usually, the name of the parent element will appear within the name of the child element as a result of these naming guidelines. For example, "Manager Employee Identifier" could be the name of a subclass of the element "Employee Identifier".

Because all occurrences of the child are by definition occurrences of the parent, the occurrences of the child element must meet all the validation requirements of the parent. This includes the rules specified in the integrity narrative, or the code set of the parent. The definition of the child may add rules to further restrict the set of valid occurrences, but does not need to repeat the rules in the narratives of the definition of its parent. However, if the parent linked to a code set, then the child must be directly linked to a code set.

notes

The element that refers to a subclass is called the child element; the element that defines the broader class is called the parent element. The occurrences of the children are, by definition, occurrences of the parent. For example, the element *Component Part Number* must be defined as a subclass of *Part Number* if all component part numbers are themselves part numbers.

The child element does not have a *Data Form* of its own. Instead, it adopts the data form of its parent. If there is a tree of subclasses, formed because a parent is simultaneously a child of another element, then the tree must eventually terminate in an element that has a data form; this is called the *root element*.

The *subclass of* relationship establishes a controlled link among those elements that have the same data form. In order to share a data form, a set of elements must share a common root element.

The *subclass of* relationship is not, strictly speaking, a hierarchical, subtyping relationship. Strict hierarchies require that every occurrence of the parent be an occurrence of only one of the child types. However, among elements, the *subclass of* relationship does not impose the requirements of exclusive or exhaustive partitioning; an occurrence of the parent element may be an occurrence of any number of the child elements, or of none.

The majority of the elements that are a subclass of another will appear only on an optimized data model, where they will reflect the denormalization process in the form of foreign keys or other transformations of relationships.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

related names

related name	context
Based On	Common Usage

SUBSET OF

business name
Subset Of

entities associated and cardinalities

Code Set	0,N
Code Set	0,N

description
The linkage of a particular code set with a composite code set that contains it as a subset.

data object type
Complex Relationship

abbreviated name
SUBSET_OF

reverse relationship name
Contains

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Whenever a code set is a subset of another, the elements that are bounded by either code set must themselves be related to a common root element through the *subclass of* relationship.

notes
Code Sets are often broken down into subsets. For example, the codes for "U.S. State Codes" could be broken down into "New England States", "Mid-Atlantic States", and so on. To avoid redundancy and ensure consistency, the larger set can be defined to contain the smaller sets. In this manner, the larger set is updated whenever the subset is updated.

There is no requirement for a strict hierarchy in this relationship. Two subsets can have overlapping values. For example, the "Northeast Region" could contain all the values from "New England States" plus half of "Mid-Atlantic States"; all these could be subsets of "U.S. State Codes."

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

TRANSFORMS INTO

business name
Transforms Into

entities associated and cardinalities

Business Data Object	0,N
Business Data Object	0,N

attributes

Transformation Narrative

description
An association between two data objects that indicates that one was created through the transformation of the data definition of the other into a new business context or level of abstraction.

data object type
Complex Relationship

abbreviated name
TRNSFMS_INTO

reverse relationship name
Was Derived From

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Because transformation is not a mechanical process, there must be a human review of any changes to the original data object, to determine the effect of those changes on the transformed object. For example, a change to a 3NF model will probably impact any optimized model that was based upon it, but the nature of this impact cannot be determined with review. Whenever the original object is replaced and becomes obsolete, the transformed object becomes invalid over time.

notes
A transformation occurs when an existing definition is modified or converted into a new definition.

A transformed definition defines the same object that the original definition defined, but under new circumstances. A new model that borrows heavily from another model in another business area is a transformation, as would be the adaptation of a very generic model to a local business. Transformation must be used in the following circumstances.

- An Optimized Model must be a transformation of a 3NF Model.
- Complex relationships must be transformed into a set of simple relationships and/or optimized entities when a 3NF model is optimized.

A Transformation may also be defined in other appropriate circumstances such as the following:

- An optimized entity is created by modifying one or more 3NF entities; the 3NF entity is transformed into the optimized entity
- An element is used as a foreign key to implement relationship; the relationship is transformed into the element.

The *transformation* relationship implies that there is a relationship between two data object definitions, but that this relationship is not a strict copy or derivation. Rather, an individual's judgment was used when the new definition was created. Subsequently, if the original definition is changed, judgment is required to determine what changes ought to be applied to the transformed definition.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

VALUES BOUNDED BY

business name
Values Bounded By

entities associated and cardinalities

Element	0,1
Code Set	0,N

description
The link between an element and the code set that establishes the valid values of that element.

data object type
Simple Relationship

abbreviated name
VAL_BNDED_BY

reverse relationship name
Defines Valid Values of

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
An element can have only one code set that establishes its valid values. This code set does not change from model to model, or from application context to application context. If there is a need to restrict the values of the code set in a given context, then a new element must be defined as a subclass of the original element. The new element can then refer to a subset of the original code set.

Whenever two elements relate to the same code set, they must be related to a common ancestor through the *subclass of* relationship. The controller of this common ancestor element is the controller of the code set.

notes
A code set establishes the specific set of values that an element can have, and optionally defines the meaning of those values. Additionally, the definition of an element contains an *integrity narrative* that establishes other rules governing the valid values of an element, including such options as upper and lower limits. The integrity narrative of the element can also note the code set used to validate values and provide examples or explanations of this code set.

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

5 STANDARD ELEMENT DEFINITIONS

This chapter contains the definitions of the elements that appear as keys or attributes of entities, or as attributes of relationships, in the two models in Chapter 2, *Standard Data Model Requirements* and *Type Hierarchy of Data Objects*.

Each of these elements represents some item of information that is related to the complete definition of a data object.

ABBREVIATED NAME

business name
Abbreviated Name

description
The abbreviated name of a business data object.

data object type
Element

abbreviated name
ABBRD_NAME

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The requirements and procedures for creating abbreviated names are specified in the DDNG. The abbreviated name is created by using the standard abbreviations of all the business terms found in the business name of the data object, as those abbreviations are found in the Business Term Glossary. Further abbreviation of the result is allowed if the original is longer than the 31-character maximum size.

The underscore character (_) is used to delimit the abbreviated business terms. Other punctuation is allowed if it appears as part of the business term abbreviation itself.

notes
The abbreviated name of an element serves as the default field name for the Standard Fields that implement it. If the DDL of the Standard Field permits it, the Standard Field must accept this default name.

derivation narrative
The abbreviated name is created from the business name using the procedures defined in the DDNG.

default action narrative
None.

element edit string
X(31)

data format type
Text

maximum character count
31

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

AGGREGATION NARRATIVE

business name
Aggregation Narrative

description
An explanation of the unique aspects of an aggregation.

data object type
Element

abbreviated name
AGRGT_NRTV

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The aggregation narrative is a free text block. It may be left blank.

notes
The *aggregation narrative* provides a record of additional business rules regarding the breakdown of one data object into others.

For example, an entity can be broken down into segments, with each segment being independently defined as an entity. The rules for the use of each segment and where it appears in the parent entity is captured in the aggregation narrative that exists on the *aggregate of* relationship between the parent entity and that segment.

derivation narrative
None

default action narrative
The missing value is "None."

element edit string
T(60)

data format type
Text

maximum character count
64K

variable size indicator
Variable

text justification

Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

BUSINESS NAME

business name
Business Name

description
The full, formal name by which a business data object can be unambiguously identified within a business.

data object type
Element

abbreviated name
BUSNS_NAME

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The business name must be created according to the requirements and procedures specified in the DDNG. All of the significant business terms (or phrases) used in a business name must be defined in the Business Term Glossary.

The business name must be unique, with the following exceptions. Two objects of different data object types can have the same business name; for example, there can be a 3NF model called *Employee* and a 3NF entity called *Employee*, but there cannot by two 3NF entities called *Employee*. The other exception is that the business names of associations are not required to be unique. For example, there can be two relationships named *consists of*, one a model. However, two associations with the same name must associate different entities.

The business name of an element should end in a business term that indicates a *Value Type*. Refer to the DDNG for specific information about value types and how they are applied.

notes
The business name is a complete, formal name of a business data object. This is the name by which the business data object should be known in all data management contexts, including implemented information systems.

The business name of an element is used as the default value of the field query name of any fields that implement this element.

derivation narrative
The business name is created from the description, using the procedures defined in the DDNG.

default action narrative
None.

element edit string
X(80)

data format type
Text

maximum character count
80

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

BUSINESS TERM

business name
Business Term

description
A term or phrase that has meaning within a business: the words that appear in business names excluding prepositions and auxiliary words. A business term can be a word, a short phrase, or an acronym.

data object type
Element

abbreviated name
BUSNS_TERM

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The DDNG defines the requirements and procedures for identifying business terms. Refer to that document for complete information.

notes
Business terms are typically single words. Business terms may be abbreviations, acronyms, or phrases, when these have strong recognition or independent meaning. For example, *USA* could be a business term, as could *Credit Card*.

derivation narrative
None

default action narrative
None

element edit string
X(31)

data format type
Text

maximum character count
31

variable size indicator
Fixed

text justification

Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

related names

related name	context
Keyword	DEC STD 065 Rev C

BUSINESS TERM ABBREVIATION

business name
Business Term Abbreviation

description
The standard abbreviation of a business term.

data object type
Element

abbreviated name
BUSNS_TERM_ABBR

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The requirements and procedures for creating business term abbreviations are defined in the DDNG.

As stated in the DDNG, cross-referencing for synonyms and homonyms of a business term must be performed before a business term abbreviation can be created.

Business term abbreviations cannot contain underscores.

notes
The business term abbreviation is used in the formulation of an abbreviated name. Refer to the DDNG for specific requirements and procedures.

derivation narrative
Business term abbreviations are formed by removing the least significant letters from business terms. The specific process is defined in the DDNG.

default action narrative
None

element edit string
X(31)

data format type
Text

maximum character count
31

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

related names

related name	context
Keyword Abbreviation	DEC STD 065 Rev C

BUSINESS TERM DEFINITION

business name
Business Term Definition

description
The formal definition of a Business Term.

data object type
Element

abbreviated name
BUSNS_TERM_DFNTN

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The specific requirements and procedures for creating business term definitions are defined in the DDNG.

notes
The definition of a business term explains what the term means to the business.

When a business term has more than one distinct definition, then the term must have distinct abbreviations. This situation is called a homonym (where one word has two meanings). When two different terms have the same definition, the terms must have the same abbreviation. This situation is called a synonym (where two words have the same meaning). Refer to the DDNG for complete information on synonyms and homonyms, and how they are managed in the Business Term Glossary.

derivation narrative
None

default action narrative
None

element edit string
T(60)

data format type
Text

maximum character count
64K

variable size indicator
Variable

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

related names

related name	context
Keyword Definition	DEC STD 065 Rev C

CODE SET NAME

business name
Code Set Name

description
The complete name by which the code set is unambiguously identified.

data object type
Element

abbreviated name
CD_SET_NAME

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The code set name must be unique.

notes
Any alternate name for a code set should be identified in the form of a *related name* of the element.

derivation narrative
The formal name of a code set is typically formed by removing the value-type term and adding the word *table* to the end of the element's business name. For example, the code set against which the element *Customer Address Type Code* is validated might be called *Customer Address Type Table*.

default action narrative
The initial value can be defaulted as described in the derivation narrative above.

element edit string
X(80)

data format type
Text

maximum character count
80

variable size indicator
Fixed

text justification

Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

CODE DESCRIPTION

business name
Code Description

description
A brief explanation of the meaning of a code value.

data object type
Element

abbreviated name
CD_DSCRPN

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The description of a code can be changed by the controller of the element that the code set applies to. If a code applies to more than one element then there is a requirement that the elements share a common root element through the *subclass of* relationship (see its definition in Chapter 4). The controller at the highest level of commonality is responsible for the values and descriptions of codes.

The code description can be left blank for an entire code set if the individual code values are self-explanatory.

notes
The code description is generally the full term for which the code itself is an abbreviation. For example, the code "F" might have the the description "Female." In some instances, the code description is used to provide more detail than a simple expansion of the abbreviation. For example, the code "DE8" on the Product Category Table might have the description, "Synchronous Interface (Character Oriented), No Control".

derivation narrative
None

default action narrative
None

element edit string
X(80)

data format type
Text

maximum character count
80

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

CODE VALUE

business name
Code Value

description
One valid value of an element that has a unique and significant business interpretation.

data object type
Element

abbreviated name
CD_VAL

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
A code value can be added, changed, or deleted only by the controller of the element that the code set applies to. If a code applies to more than one element then there is a requirement that the elements share a common root element through the *subclass of* relationship (see its definition in Chapter 4). The controller at the highest level of commonality is responsible for the values and meanings of codes.

A code value cannot exist independently of a code set; and a code set cannot exist independently from an element. The element to which the set is connected provides the data form of the code value; each code value is an occurrence of the element.

notes
The code value is one valid value of an element. Each code value must be unique for the element it applies to and must have a predefined significance to the business.

The element *code value* becomes a subclass of whatever element it is associated with. Each instance of the code value must adhere to the data form of the element being bounded.

derivation narrative
None

default action narrative
None

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

COMPUTED-BY EXPRESSION

business name
Computed-By Expression

description
The DDL statement that produces the values for a field; the language-specific implementation of the algorithm for the derivation or computation of the values.

data object type
Element

abbreviated name
CMPUTD-BY_XPRSN

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The derivation narrative of an element states the business rules for calculating the values of an element. The computed-by expression must be an implementation of its associated derivation narrative.

The expression must be provided in a format acceptable to the host DDL, and must be executable at runtime in the application environment.

The computed-by expression may contain formatting characters such as Tab and Carriage Return, subject to the rules of the host language.

The format and requirements for expressions in CDO are defined in *VAX CDD/Plus Common Dictionary Operator Reference Manual*.

notes
In CDO, the computed-by expression can contain value expressions or conditional expressions. Value expressions can be composed of the following:

- Fields or records
- Literals
- Statistical expressions (average, maximum, and so forth)
- Arithmetic expressions (add, subtract, and so forth)

- Concatenated expressions
- First-from expressions
- Table expressions

Examples of value expressions are:

```
UNIT_PRICE * QTY
COUNT OF E IN EMPLYE
```

Conditional expressions consist of value expressions, relational operators (equal, greater, not equal), or logical operators (and, or, not). For example:

```
IF QTY LT 100
    THEN UNIT_PRICE * QTY
    ELSE (UNIT_PRICE * .95) * QTY
```

derivation narrative
None

default action narrative
None

element edit string
X(256)

data format type
Text

maximum character count
256

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

COMPUTED-BY ROUTINE NAME

business name
Computed-By Routine Name

description
The logical name or pointer to a subroutine that produces the values for a field.

data object type
Element

abbreviated name
CMPUTD-BY_ROUTN_NAME

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The name must be provided in a format acceptable to the host DDL. The routine must be executable at runtime in the application environment (using compatible runtime libraries, for example).

notes
This name must refer, at runtime, to an available subroutine or callable service. The routine name can be a DNS name, a symbol in the runtime library, an installed image, a subroutine in the program image, or any other identification as supported by the application environment.

derivation narrative
None

default action narrative
None

element edit string
X(31)

data format type
Text

maximum character count
31

variable size indicator
Fixed

text justification

Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

CONTROLLER NAME

business name
Controller Name

description
The unique name that identifies a controller.

data object type
Element

abbreviated name
CNTRLR_NAME

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
A controller can be a single job role or job title, a committee, an organization, or a position; the controller name must be the name of one of these. The controller name must not be an individual person's name.

notes
The name of the controller must be sufficient to identify that agent or agency.

derivation narrative
None

default action narrative
None

element edit string
X(60)

data format type
Text

maximum character count
60

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

CONTROLLER CONTACT EMPLOYEE IDENTIFIER

business name
Controller Contact Employee Identifier

description
The corporate employee identifier of an individual designated by the controller as the contact point for questions or issues regarding business data objects.

data object type
Element

abbreviated name
CNTRLR_CNTCT_EMPLYE_ID

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The contact point must be an employee of the company, and is identified by their corporate employee identifier.

The controller may change their contact point at any time, without requiring reapproval or a new revision of any definitions.

notes
The controller contact employee identifier is used to identify an individual employee who is the contact point for questions and issues regarding the definitions of business data objects. This individual is not necessarily empowered to make decisions on behalf of the controller, but would be expected to be knowledgeable in the data domains over which the controller has authority.

If the controller is a single individual (identified though their job title or position), they may identify themselves as the contact point.

Any time a contact employee changes job, organization, or role, it is up to the controller to determine whether a new contact employee should be identified.

derivation narrative
None

default action narrative
None

subclass of

Employee Corporate Badge Number

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

DATA DESCRIPTION LANGUAGE NAME

business name
Data Description Language Name

description
The formal name of a language used for defining implementation data objects.

data object type
Element

abbreviated name
DDL_NAME

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The DDL Name is specified at the time that the first field in this DDL is created.

notes
The *Data Description Language Name* can refer to a family of DDLs or to an individual language.

The DDL name *CDO* refers to the language of the VAX CDD/Plus Common Dictionary Operator utility.

derivation narrative
None

default action narrative
The default initial value is *CDO*.

element edit string
X(30)

data format type
Text

maximum character count
30

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

DATA FORMAT TYPE

business name
Data Format Type

description
The general class of the format of the data represented by an element.

data object type
Element

abbreviated name
DATA_FRMT_TYPE

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The data format type must be one of the values from the following table.

Table 5–1 Data Format Type Table

data format type	data format type description
Text	The element contains character data.
Numeric	The element contains computational data.
Date	The element specifies a date and/or time.

notes
The entity *data format* has three subtypes. The element *data format type* is the key by which these three types of data format are distinguished.

Only three types of data can currently be standardized on data models: *Text, Date*, and *Numeric*. Other data formats exist, such as video, compound document, and voice. These formats are not included in the current version of this definition, but may be added in future revisions. Alternatively, these may be handled by using an existing data form. For example, a compound document could be represented by a textual filename.

derivation narrative
None

default action narrative
The default initial value is *Text* because it is the most common data format.

element edit string
X(8)

data format type
Text

maximum character count
8

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

DATA OBJECT IDENTIFIER

business name
Data Object Identifier

description
The unique identifier of a data object; this identifier has no significance borrowed from the meaning of the data object, from the connection of this object with other data objects, or from the point of origin of the definition.

data object type
Element

abbreviated name
DATA_OBJCT_ID

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The data object identifier does not change for the life of the object. The data object identifier is simply a character string that is assigned to a data object at the time it is created. The string does not specify where it was created, its status, the type of data object, or any other factor related to the management of the object; such requirements must be met by a distinct attribute of the object. The identification of the data object must be unique across all physical, organizational, procedural, and implementation boundaries.

notes
The data object identifier is part of the primary key of the entity, Data Object. The other portion of the key is the data object revision.

The data object identifier is not suitable for presentation to users as meaningful data; its primary role is to facilitate unique reference to an object, independent of changes to the name or the revision of the object.

In general usage, the business name of data object is used to refer to that object. However, over time there may be a requirement to change the name of that data object. Thus the data object identifier is established independently, to support the requirement for continuity despite changes to the name. Note also that DEC STD 065-0 does not prohibit duplicate business names. For example, the names of relationships such as *Consists of* or *Rolls Up to* could be reused for many different relationships.

derivation narrative
None

default action narrative
None

element edit string
X(10)

data format type
Text

maximum character count
10

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

DATA OBJECT REVISION

business name
Data Object Revision

description
A sequential number used to indicate the revision of the data object definition relative to other revisions.

data object type
Element

abbreviated name
DATA_OBJCT_RVSN

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The data object revision is a sequential number; it begins at 1 when the data object is created and is incremented sequentially.

notes
A revision of a data object occurs when a data object definition is altered but the new version is designed to be used wherever the previous version was used. When this happens, the data object identifier remains unchanged and the data object revision is incremented by 1.

The data object revision is incremented when the alterations begin, while at the same time the status is set to *Work in Process*. The revision does not need to be incremented if a definition is altered while its status is *Work in Process*.

When a new revision of a data object definition is approved, it automatically replaces the previous revision (see the definition of the relationship, *replaces*), and the status of the previous revision is set to *Obsolete*.

derivation narrative
The revision of a newly created version of a data object can be established by adding 1 to the value of the previous revision.

default action narrative
The default revision for newly created data objects is 1.

element edit string
ZZ9

data format type
Numeric

size in digits
3

element scale
0

signed numeric indicator
Unsigned

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

DATA OBJECT STATUS

business name
Data Object Status

description
The stage in the definitional process of this data definition.

data object type
Element

abbreviated name
DATA_OBJCT_STATUS

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The data object status must be one of the values listed in the following table.

Table 5–2 Data Object Status Table

status values	status meaning
Work In Process	Definition is subject to change without notice
Approved	Definition can be used in any model
Obsolete	Definition cannot be added to new models

notes
When a data object is initially created, its data object status must be set to *Work in Process*. If a new revision of an existing object is being created, the status of the new revision must begin as *Work in Process*. Note that while the new revision is being worked on, the prior revision remains in effect.

A data object's status may change to *Approved* when it has met the requirements established by the Enterprise Data Board, under their authority described in Section 1.4.1. At a minimum, a data object definition must meet the requirements stated in this standard in order to become approved.

A data object becomes *Obsolete* when a new revision is approved, when a new data object is defined that is intended to replace this data object, or when the data object is no longer of interest to the business. An obsolete definition cannot be added to new models, but it can continue to be called

out on existing models until those models are themselves revised. When there are no longer any references to an obsolete data object, it can be archived.

A data object may become *Obsolete* at the discretion of the controller when there is no replacement data object defined. In other words, a controller may inactivate a definition.

derivation narrative
None

default action narrative
The initial value of the data object status is always *Work in Process*.

element edit string
X(18)

data format type
Text

maximum character count
18

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

DATA OBJECT TYPE

business name
Data Object Type

description
The specific type of data object.

data object type
Element

abbreviated name
DATA_OBJCT_TYPE

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The data object type must be one of the values listed in the following table.

Table 5–3 Data Object Type Table

type values	type meaning
3NF Model	Data model in third normal form
Optimized Model	Data model optimized for implementation
3NF Entity	Data entity in third normal form
Optimized Entity	Data entity optimized for implementation
Classification	Specialization of an entity into categories
Subtyping	Specialization of an entity into subtypes
Simple Relationship	Binary, functional relation between entities
Complex Relationship	Complex relationship among entities
Element	Data item of interest to the business
Standard Field	The global definition of the implementation of an Element within a particular DDL
Nonstandard Field	A locally modified version of a Standard Field

notes
The data object type determines the identity of the data object within the model, *Type Hierarchy of Data Models*.

derivation narrative
None

default action narrative
None

element edit string
X(20)

data format type
Text

maximum character count
20

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

DEFAULT ACTION NARRATIVE

business name
Default Action Narrative

description
The description of the actions that should be taken when a value of an element is not supplied by a user or not present in a data base.

data object type
Element

abbreviated name
DFLT_ACTN_NRTV

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The default action narrative is a free text block. It may be left blank.

notes
The default action narrative contains all the information that is needed to understand or implement the actions that are taken when the value of a data item is not explicitly established. Three types of defaults can be described in this narrative: Initial Value, Missing Value, and Antecedent Value.

An *Initial Value* is the value automatically assigned when the data item is created, unless a different value is supplied at that time. This value is most commonly called the default. For example, the initial value is sometimes prefilled on a data entry screen and is stored in a file unless the data entry operator provides a different value. Once it is created and stored, an Initial Value is indistinguishable from an explicitly provided value.

A *Missing Value* is the value that is used in operations when a value for a data item has not been recorded. This differs from an Initial Value, because an initial value is actually stored in a data base whereas the Missing Value is used when, for instance, there is no record at all. For example, the element *Spouse Name* might have a missing value of *none*. The display of information about an employee would show *none* as the spouse name for an unmarried employee. However, if the data base were searched for all *Spouse Names* of *none*, no employee records would be found. Missing values are especially important for numeric elements that will be subject to statistical expressions (such as averaging, or maximum /minimum calculations). Because a missing value is not actually stored, these values will not be used in the calculations. For example, if the

missing value of the element *Part Failure Rate* is 0, this will not affect the calculation of the *average failure rate, total failure rate*, or *minimum failure rate*. If on the other hand the Initial Value had been 0, these values would have impacted the calculations.

An *Antecedent Value* is the value that should be assigned when an element is added to a pre-existing entity. For example, if the element *Country of Citizenship* is added to the *Employee* entity, there could be instructions that state that all existing *Employees* should have the *Country of Citizenship* initialized to *Unknown*. This value is often not the same as the Initial Value that is used for newly recorded entities, which in this example might be the name of the country where the employee is being hired.

derivation narrative
None

default action narrative
The missing value is "None."

element edit string
T(60)

data format type
Text

maximum character count
64K

variable size indicator
Variable

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

DERIVATION NARRATIVE

business name
Derivation Narrative

description
A statement of the business rules for the derivation or computation that produces the values of the element.

data object type
Element

abbreviated name
DERVN_NRTV

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The derivation narrative is a free text block. It may be left blank.

notes
The derivation narrative provides all the information needed to understand or implement a data item in which values are based on some function or combination of other elements. The rules must refer to the underlying data elements explicitly by name.

A derivation narrative must be supplied for any data element in which values are based on the values of other data elements, even if they are not dynamically computed. For example, an element such as *Product Cost* might be a fixed value, but it would also be based on a set of business rules for its computation. These business rules are recorded in the derivation narrative even though the computation is a business procedure rather than a programmed procedure.

The rules for normalization require the removal of any elements that depend on other elements for their value. This generally prohibits any derived element from appearing on a 3NF entity or 3NF model. However, if the derivation is not algorithmic then the element can be included on the 3NF model.

derivation narrative
None

default action narrative
The missing value is "None."

element edit string
T(60)

data format type
Text

maximum character count
64K

variable size indicator
Variable

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

DESCRIPTION

business name
Description

description
A succinct, context-free description of a data object.

data object type
Element

abbreviated name
DSCRPN

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The requirements and procedures for producing a description for a business data object are defined in the DDNG. All descriptions must conform to these requirements.

notes
The DDNG defines the rules and procedures for describing business data objects. In summary, the description is developed without regard for the existing name of the object, and it focuses on precisely *what* the data object is. The description must avoid reference to who uses the object, why it is used, where it is used, or the application that uses it.

The description of a business data object should always avoid dependence on the particular context in which it is first defined, so that it can be reused meaningfully in other context later on.

The description becomes the source for the business terms that are used in the business name. These terms must be used, in the description, with the meanings defined in the Business Term Glossary. Refer to the DDNG for further information.

derivation narrative
None

default action narrative
None

element edit string
T(60)

data format type
Text

maximum character count
64K

variable size indicator
Variable

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

DOCUMENTATION SOURCE CITATION

business name
Documentation Source Citation

description
The full bibliographic citation of a published reference source.

abbreviated name
DOCMT_SRC_CITATN

data object type
Element

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Bibliographic citations are composed of a set of formal pieces of information. At a minimum, a documentation source citation must identify the name of the document, the publisher, the edition and date, and the author, if there is one.

notes
This element represents the full bibliographic citation of a reference document. It is used to provide a common reference point for all participants in a technical field.

derivation narrative
None

default action narrative
None

element edit string
T(60)

data format type
Text

maximum character count
64K

variable size indicator
Variable

text justification

Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

ELEMENT EDIT STRING

business name
Element Edit String

description
The rules for formatting the element upon display.

data object type
Element

abbreviated name
ELEMT_EDITSTR

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The element edit string must be composed using the CDO definition language. The complete syntax for edit strings in CDO can be found in the *VAX CDD/Plus CDO Reference Manual*.

notes
An element edit string must be written in the CDO language of CDD/Plus. When a field edit string is created, it must conform to the language rules of the host DDL for that field.

Previous editions of DEC STD 065 differentiated among a Display Format, an Input Format, and an Interchange Format, all of which were edit strings. This differentiation is no longer necessary; the standard Interchange Format is now replaced by the element edit string. Data in DDLs other than CDO and data from nonstandard fields must be converted to this specified format whenever it is shared among applications.

Any special requirements for input formatting must be described in the element's integrity narrative, and may be implemented in either or both of the field edit string and input validation expression or routine that are associated with the field.

derivation narrative
None

default action narrative
The default element edit string can be derived from the data form of the element. If the data form is *Text*, the edit character X with a repeat count equal to the *Size in Characters* would provide an edit string. If the data form is *Numeric*, the edit character 9 with a repeat count equal to the

Size in Digits, with a decimal point edit string inserted according to the *Element Scale*, would produce an edit string. If the data form is *Date*, then "dd-mmm-yyyy rr:pp:ss" is the default edit string.

element edit string
X(30)

data format type
Text

maximum character count
30

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

ELEMENT SCALE

business name
Element Scale

description
The degree of precision of a numeric value; the position of the least significant digit relative to the decimal point. A positive scale indicates that the precision is to left of the decimal point, and a negative scale indicates that the precision is to the right of the decimal point.

data object type
Element

abbreviated name
ELEMT_SCALE

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The element scale must be a signed integer. The value must be less than the Maximum Character Count of the element it describes.

notes
The element scale defines the degree of precision of a numeric value. The element scale can be interpreted as stating, "all of the number positions to the right of this point should be displayed as 0s".

The element scale does not replace or supersede the size or edit string of an element. The size in digits of the element determines the largest value that can be stored; and the element edit string determines the presentation format and decimal alignment. Typically (depending in part on the implementation language), a data value is stored according to the limits of the edit string rather than the scale.

An element scale is most relevant if the element represents a physical measurement. For elements representing physical measurement, the scale is often determined by the gauge used to perform the measurement. For example, the element Automobile Mileage could have an element scale of -1 because most automobile odometers measure mileage to the nearest tenth-of-a-mile.

The element scale may also be used to round-off (or truncate) an irrelevant level of detail. For example, the element Reimbursable Personal Car Mileage might have a scale of 0 because in this context there may be no relevance to the tenths-of-a-mile detail. Commonly, summary financial reports will present dollar values with an element scale of three 3 which

means, rounded off to the nearest $1,000. Note that the implementation of rounding versus truncation of nonsignificant digits is controlled by the implementation language.

derivation narrative
None

default action narrative
The initial value is 0.

element edit string
SS9

data format type
Numeric

maximum character count
3

element scale
0

signed numeric indicator
Signed

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

FIELD DATATYPE

business name
Field Datatype

description
A specific type of storage formatting, as supported by a DDL.

data object type
Element

abbreviated name
FLD_DTYPE

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The field datatype must be a valid datatype in the host DDL.

For standard fields in CDO, the field datatype must be consistent with the data format type of the element being defined.

If the DDL does not require a datatype (or, if it is declared in the DDL edit string or elsewhere), then the field datatype may be left blank.

notes
The field datatype indicates the form of allowable data, and for numeric datatypes the number of available digits.

derivation narrative
None

default action narrative
The default initial value of a field datatype is drawn from the data format type of the element as follows. If the data format type is *Text*, the field datatype is TEXT. If the data format type is *Date*, the field datatype is DATE. If the data format type is *Numeric*, the field datatype is either SIGNED NUMERIC or UNSIGNED NUMERIC depending on the element Signed Numeric Indicator.

element edit string
X(30)

data format type
Text

maximum character count
30

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

FIELD EDIT STRING

business name
Field Edit String

description
The rules for formatting a field upon display.

data object type
Element

abbreviated name
FLD_EDITSTR

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The field edit string must be valid in the field DDL.

notes
The field edit string must be used to provide the host DDL with sufficient specification of the field to support consistent implementation across all applications that use that field. If the DDL determines the formatting from other parameters that are not specifically called out in this data model, then the specification of these other items should be embedded in the field edit string.

derivation narrative
If the DDL of a standard field is CDO, then the field edit string must match the element edit string.

default action narrative
For standard fields, the initial value is copied from the element edit string of the defined element. For nonstandard fields, the initial value is copied from the field edit string of the standard field on which the nonstandard field is based.

element edit string
X(256)

data format type
Text

maximum character count
256

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

FIELD HELP

business name
Field Help

description
The standard help message for a field.

data object type
Element

abbreviated name
FLD_HELP

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
If field-help is provided through the facilities of a dictionary in the host DDL, the field help must comply with the requirements of that DDL.

The field help of a standard field should be as independent of application context as possible, so that the standard field can be reused in other contexts.

notes
If the host DDL provides the capability of specifying field-help, this is the message used in that context. If not, this message must be made available through whatever procedure or help facility is applicable.

derivation narrative
The field help should mirror the description of the element being implemented, subject to restrictions of length.

default action narrative
None

element edit string
X(256)

data format type
Text

maximum character count
256

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

FIELD JUSTIFICATION

business name
Field Justification

description
The rules for the alignment of data within a field.

data object type
Element

abbreviated name
FLD_JSTFTN

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The field justification can only be used if the host DDL supports user-specified justification for fields, and for the datatype of this field. The host DDL determines the valid values for this attribute, and how they are to be interpreted.

If the DDL of a standard field is CDO, the field justification must match the text justification of the element being defined.

The valid values in CDO are *Left, Right*, and *Center*. In CDO, the justification can only be specified for fields with datatype of *Text* or *Unspecified*. The alignment of numeric data is accomplished within the specification of the datatype itself.

notes
The field justification is available in some DDLs as a means to control the alignment of data within a field when fewer than the maximum number of characters is provided. Depending on the host DDL, the field justification may also dictate which end of the data is truncated if a value longer than the maximum character count is moved into this field. Refer to the documentation of the host DDL for complete information.

derivation narrative
The field justification for a standard field takes as its initial default value the text justification of the element being defined. This can then be modified to conform to the syntactic requirements of the host DDL. If the element has a data form other than *Text*, the field justification cannot be derived.

The field justification for a nonstandard field takes as its initial default value, the field justification from the standard field upon which this nonstandard field is based. This may then be modified; refer to the definition of *Nonstandard Field* in Chapter 3 and the definition of *Based On* in Chapter 4 for additional guidelines.

default action narrative

The initial value for standard fields is copied from the text justification of a textual element data form; if the element does not have a *Text* data form, then there is no default value for the field justification of a standard field. The initial value for nonstandard fields is copied from the field justification of the standard field it is based on.

element edit string

X(16)

data format type

Text

maximum character count

16

variable size indicator

Fixed

text justification

Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

FIELD LABEL

business name
Field Label

description
The label used to identify a field on reports, screen displays, and data collection prompts.

data object type
Element

abbreviated name
FLD_LABEL

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
If the host DDL supports the labeling of fields within the data definition, then the field label must conform to the syntactical requirements of the DDL. If the host DDL does not support dictionary-driven labeling, then the procedural language that builds the report (or screen display) must be programmed to use this label; and the label should conform to the requirements of the application procedural language.

notes
The field label is the label that should be used to identify this field in any presentation context, such as on reports, on screen displays, or as a prompt.

The label for an element should be unique within an application; and a single element should have only one label within an application. Variation of the label from application to application should be avoided in order to promote consistency and sharability.

If business conditions dictate a need for a different label, then a nonstandard field definition must be created so that the correct connection of the nonstandard label back to the original business element is preserved.

derivation narrative
The field label for a standard field takes as its initial default value, the business name of the element being defined. This can then be modified to conform to the syntactic requirements of the host DDL, and be abbreviated to a reasonable size.

The field label for a nonstandard field takes as its initial default value, the field label from the standard field upon which this nonstandard field is based. This may then be modified; refer to the definition of *Nonstandard Field* in Chapter 3 and the definition of *Based On* in Chapter 4 for additional guidelines.

default action narrative

The initial value for standard fields is copied from the business name of the element; the initial value for nonstandard fields is copied from the field label of the standard field it is based on.

element edit string

X(80)

data format type

Text

maximum character count

80

variable size indicator

Fixed

text justification

Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

FIELD NAME

business name
Field Name

description
The identifying name of a field; this is the name that applications must use to refer to the field.

data object type
Element

abbreviated name
FLD_NAME

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The field name must conform to the requirements of the host DDL.

The field name of a standard field must match the abbreviated name of the element, unless prohibited by the requirements of the host DDL.

The complete requirements and procedures for creating field names for standard fields are defined in the DDNG.

The name of a field of one element cannot be identical to a name of a field for a different element. However, two fields for the same element may have the same field name.

notes
The field name is the name used by applications to refer to a field. The standard field name is borrowed from the abbreviated name of the element that is being implemented.

When a modified field name is required for processing purposes, a nonstandard field must be created with the new name. However, when a modified name is required to convey a different business connotation, then this difference must be reflected as a new element that will then be added to the optimized model upon which this implementation environment is based.

derivation narrative
The field name for a standard field takes as its initial default value, the abbreviated name of the element being defined. This can then be modified to conform to the syntactic requirements of the host DDL.

The field name for a nonstandard field takes as its initial default value, the field name from the standard field upon which this nonstandard field is based. This may then be modified; refer to the definition of *Nonstandard Field* in Chapter 3 and the definition of *Based On* in Chapter 4 for additional guidelines.

default action narrative

The initial value for standard fields is copied from the abbreviated name of the element; the initial value for nonstandard fields is copied from the field name of the standard field it is based on.

element edit string

X(31)

data format type

Text

maximum character count

31

variable size indicator

Fixed

text justification

Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

FIELD QUERY NAME

business name
Field Query Name

description
A name for a field that can be used in interactive queries.

data object type
Element

abbreviated name
FLD_QRY_NAME

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The field query name must conform to the requirements of the host DDL. If the host DDL does not support Query Names, this field should be left blank.

notes
The field query name provides an alternate name for a field that is usable through interactive query languages. Both the DDL and the query language must support this construct in order to make use of the field query name.

The purpose of a query name in query languages is to provide end users with a name that is not as obscure as most field names, corresponds to the terms used for information, and that is easy to type. All these requirements should be balanced when creating a good field query name.

derivation narrative
The field query name for a standard field takes as its initial default value the business name of the element being defined. This can then be modified to conform to the syntactic requirements of the host DDL.

The field query name for a nonstandard field takes as its initial default value the field query name from the standard field upon which this nonstandard field is based. This may then be modified; refer to the definition of *Nonstandard Field* in Chapter 3 and the definition of *Based On* in Chapter 4 for additional guidelines.

default action narrative
The initial value for standard fields is copied from the business name of the element; the initial value for nonstandard fields is copied from the field query name of the standard field upon which it is based.

element edit string
X(80)

data format type
Text

maximum character count
80

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

FIELD SCALE

business name
Field Scale

description
The degree of precision of a numeric value; the position of the least significant digit relative to the decimal point. A positive scale indicates that the precision is to left of the decimal point, and a negative scale indicates that the precision is to the right of the decimal point.

data object type
Element

abbreviated name
FLD_SCALE

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The rules of the host DDL dictate the syntax and meaning of this element as it appears in a field definition.

notes
The field scale implements the intent of the element scale, for the element being defined. For fields created for CDO, the field scale should be equal to the element scale unless local processing requirement force a change. Refer to the definition of *Element Scale* in this chapter for information about the intent of element scale.

derivation narrative
If the DDL of a standard field is CDO, then the field scale must match the element scale of the element being defined.

default action narrative
For standard fields, the initial value is copied from the element scale of the element being defined. For nonstandard fields, the initial value is copied from the standard field defined for the same host DDL.

element edit string
SS9

data format type
Numeric

maximum character count
3

element scale
0

signed numeric indicator
Signed

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

FIELD SIZE

business name
Field Size

description
The size of the field specified for the host DDL.

data object type
Element

abbreviated name
FLD_SIZE

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The field size must be specified in a way appropriate to the datatype, according to the rules of the host DDL.

If the DDL of a standard field is CDO, the field size must match the maximum character count of the element defined, if the data form of the element is *Text*.

notes
The field size generally defines the maximum number of characters or digits that can be accepted into a field. For text data, this usually defines as well the length of the field in bytes. For numeric data, the physical size of the field is determined by the datatype selected, and the field size defines the number of actual digits that may be accepted into the field. Floating-point datatypes may not permit the specification of a field size at all, depending on the DDL; this is true also for Date datatypes.

derivation narrative
The field size for a standard field takes as its initial default value the maximum character count. This can then be modified to conform to the syntactic requirements of the host DDL.

The field size for a nonstandard field takes as its initial default value, the field size from the standard field upon which this nonstandard field is based. This may then be modified; refer to the definition of *Nonstandard Field* in Chapter 3 and the definition of *Based On* in Chapter 4 for additional guidelines.

default action narrative
The initial value for standard fields is copied from the maximum character count; the initial value for nonstandard fields is copied from the field size of the standard field upon which it is based.

element edit string
ZZZZZ9

data format type
Numeric

maximum character count
6

element scale
0

signed numeric indicator
Unsigned

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

INITIAL VALUE

business name
Initial Value

description
The value that the field is initialized to when it is created; or the default value that is to be prefilled on data entry screens when the field is being created.

data object type
Element

abbreviated name
INIT_VAL

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The initial value for a field must conform to the requirements of the host DDL for specification of initial value, if the DDL supports this feature. If the initial value will be implemented using programming procedures or forms definitions, the initial value must meet the physical requirements and integrity constraints of the field itself.

notes
The initial value is a simple constant or literal value, to which a field is initialized when it is created. This initialization can be implemented through the prefilling of the field on a data entry form, by an action within the application, or, if the host DDL supports it, through a declaration in the application dictionary.

derivation narrative
The *initial value* of a field is an implementation of the rules and requirements that are specified in the *default action narrative* and *integrity narrative* of the element being defined.

default action narrative
When no initial value is specified for a field, the initialization depends on the rules of the DDL and the processing language used to create the field. The common initial values are 0 for numeric fields, Spaces for text fields, and 17-Nov-1858 00:00 for dates.

element edit string
X(256)

data format type
Text

maximum character count
256

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

INPUT VALIDATION EXPRESSION

business name
Input Validation Expression

description
The computerlanguage expression that validates any input for form, a range of valid values, specific values, consistency checking, or any other preacceptance evaluation.

data object type
Element

abbreviated name
INP_VALDTN_XPRSN

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
If the host DDL supports an input validation expression, then it must conform to the requirements of this DDL. If the DDL does not support input validation directly, the input validation expression should be written in the programming language that will be used to create this field; the expression must eventually be copied into the procedures that prevalidate the field.

notes
The input validation expression provides a means to directly define the tests or actions performed on new values of the field, prior to being stored. If the validation takes place in some form of subroutine or callable service, then the input validation routine would be used to identify it. The two are not mutually exclusive.

derivation narrative
The input validation expression is created as an implementation of the rules and requirements in the integrity narrative of the element being defined, or the code set name that bounds the values of the element if one exists.

default action narrative
If the element being defined has a code associated with it, then the default input validation is that valid values of the field must be on that table. The particular expression used will vary from DDL to DDL, and the particular valid values may change to reflect field size or other physical parameter differences.

element edit string
T(60)

data format type
Text

maximum character count
64K

variable size indicator
Variable

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

INPUT VALIDATION ROUTINE NAME

business name
Input Validation Routine Name

description
The logical name or pointer to a subroutine that prevalidates the values for a field.

data object type
Element

abbreviated name
INP_VALDTN_ROUTN_NAME

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
If the host DDL supports the specification of an input validation routine, then this name must conform to the requirements of this DDL. If the DDL does not support direct specification of a routine name, the input validation routine should be identified in the fashion required by the programming environment from which it will be called.

notes
The input validation routine provides a means to identify a callable procedure that performs the function of prevalidating new values of the field.

This name should refer at runtime to an available subroutine or callable service. The routine name can be a DNS name, a symbol in the runtime library, an installed image, a subroutine in the program image, or any other identification as supported by the application environment.

derivation narrative
Although the name of the routine is not itself derived, the behavior of the routine must be an implementation of the integrity narrative of the element being defined.

default action narrative
none

element edit string
X(80)

data format Type
Text

maximum character count
80

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

INTEGRITY NARRATIVE

business name
Integrity Narrative

description
The explanation of the business rules governing the creation, modification, and deletion of the data object; the scope of what is included in the data object versus what is excluded; and the formal interdependencies of this data object with others.

data object type
Element

abbreviated name
INTGRTY_NRTV

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
In definitions of data models, the integrity narrative must explicitly define the scope of that model. The integrity narrative of a model should be able to answer questions such as "Why was a certain entity left out?"

The integrity narrative of an element serves as the business specification of the input validation of the fields that are defined for the element.

The integrity narrative of a relationship must specify any interdependence with other relationships. For example, if two relationships are exclusive, it would be documented in the integrity narrative of both relationships.

The integrity narrative of a classification or subtyping association must specify the criteria by which each occurrence of the parent entity is assigned to the correct child entity.

If an entity is totally dependent on another entity for its existence (typically known as a "weak entity") then this should be stated in the integrity narrative.

notes
The integrity narrative is the business specification of what is and is not included within a data object and the rules for what information must be present, what values are legal, or what conditions must exist for a data object to be modified or deleted.

The integrity narrative does not need to repeat information that is required in other aspects of the definition. For example, the cardinalities in a relationship are a form of integrity rule, as is the element edit string; these rules should be mentioned in the integrity narrative only if there is additional information beyond that already provided.

derivation narrative
None

default action narrative
The missing value is "None."

element edit string
T(60)

data format type
Text

maximum character count
64K

variable size indicator
Variable

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

MAXIMUM CARDINALITY

business name
Maximum Cardinality

description
The maximum number of times that a single occurrence of an entity can participate in a particular relationship.

data object type
Element

abbreviated name
MAX_CRDNLY

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The maximum cardinality must be an integer greater than 0, or the letter N. The N indicates *many*; that is, there is no fixed maximum.

When there is a specific number of times that each occurrence of the entity must participate in a relationship, that number will appear as both the minimum cardinality and the maximum cardinality, in the majority of cases, either 1 or N.

notes
The maximum cardinality indicates the number of occurrences of the relationship in which the entity can participate. Because all relationships join one entity with another, this indicates the maximum number of occurrences of the other entity to which the associated entity can be related.

Reading the maximum cardinalities of both of the entities in a relationship tells you if this is a one-to-one, a one-to-many, or a many-to-many relationship.

derivation narrative
None

default action narrative
The most common maximum cardinality is N, so this is the default initial value.

element edit string
X

data format type
Text

maximum character count
1

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

MAXIMUM CHARACTER COUNT

business name
Maximum Character Count

description
The maximum number of characters that can appear in a value of a data element.

data object type
Element

abbreviated name
MAX_CHAR_CNT

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The count must be an integer greater than 0.

notes
The maximum character count is the length of the largest (longest) value of an element. This size becomes the default field size for standard fields that define the element. If the size of a textual element is variable, the the variable size indicator must be set to indicate *Variable*. In this case, the field size is either equal to this largest size, or the field can be defined to have a variable length datatype. More commonly with unbounded text elements, a field is defined with the length of a single line of text and each line of the element is captured as a distinct occurrence of that field.

For numeric data, the maximum character count is used to indicate the maximum number of digits, *not* the highest actual value of the element. The count should include digit positions to the right of the decimal point, if appropriate.

For dates, this count may be omitted, or it may be used to indicate the full length of displayed data. For example, if the element edit string of a date were, DD"-"MMM"-"YYYY, the maximum character count would be 11.

derivation narrative
None

default action narrative
None

element edit string
ZZZZZ9

data format type
Numeric

maximum character count
6

element scale
0

signed numeric indicator
Unsigned

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

MINIMUM CARDINALITY

business name
Minimum Cardinality

description
The minimum number of times that an occurrence of an entity can participate in a particular relationship.

data object type
Element

abbreviated name
MIN_CRDNLY

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The minimum cardinality must be 0 or a positive integer.

In the vast majority of cases, the minimum is either 0 or 1. When there is a specific number of times that each occurrence of the entity must participate in a relationship, that number will appear as both the minimum cardinality and the maximum cardinality.

notes
The minimum cardinality identifies whether the participation of an entity in a relationship is optional or mandatory; and if mandatory, whether there is a fixed minimum number of times it must participate.

A value of 0 indicates that the participation is optional. That is, an occurrence of the entity might not be involved in this particular relationship at all. A value of 1 indicates that participation is mandatory; that is, each occurrence of the entity must participate in the relationship at least once. Any value greater than 1 also indicates mandatory participation.

derivation narrative
None

default action narrative
The most common minimum cardinality is 1, so this is the default initial value.

element edit string
X

data format type
Text

maximum character count
1

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

MISSING VALUE

business name
Missing Value

description
The value that is presumed for a field for selection, display, and transmission when the field does not have an existing value.

data object type
Element

abbreviated name
MISSG_VAL

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The missing value must be an implementation of the default action narrative of the element being defined.

The missing value of a field must conform to the requirements established by the field datatype, field size, and field edit string.

The missing value must meet the requirements of the host DDL. If the host DDL does not support a missing value construct, then this element should be omitted from the field definition.

notes
The missing value, along with the initial value, is used to implement the rules in the default action narrative of an element. The missing value is used when a field is referenced, by a computation or a selection, but there is no value for that field existing in the dataset.

This contrasts with the initial value, which is used to establish the value of the field when one is not supplied. The initial value allows an application to establish a value automatically, and to store this value in the data base. The missing value tells the application what to do when no value exists in the data base.

An example of the use a missing value might be the number of *Students Registered* for a particular *Course*. When information about the course is first created, there will have been no registrations. If an *initial value* is used in this case (for instance, the initial value is 0) then this would invalidate any computation such as the "Average of *Students Registered* for all courses." Instead, a *missing value* should be defined (rather than an initial value). The same computation will then omit any course for which there are no *Students Registered*.

The missing value is displayed whenever the field is displayed. The missing value is used in selection expressions. The missing value is not used in computations.

derivation narrative
None

default action narrative
None

element edit string
X(256)

data format type
Text

maximum character count
256

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

NOTES

business name
Notes

description
The free-form narrative containing information that amplifies the meaning of the data object. The information may include references to other sources and documents that further define the object or its occurrences.

data object type
Element

abbreviated name
NOTE

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Basically there are no limits on the content of notes. However, the information should avoid depending on a specific context, application, or data structure, because the same data object will probably appear again on other models and structures. If the information must be specific to a given context, the context should be identified.

notes
The notes provide additional information about a data object, which falls outside the formal description.

The notes should provide all the information needed to understand or implement the object, that is not already provided under the more specific headings such as integrity narrative or description.

derivation narrative
None

default action narrative
The missing value is "None."

element edit string
T(60)

data format type
Text

maximum character count
64K

variable size indicator
Variable

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

RELATED NAME

business name
Related Name

description
A name that is an alias, synonym, or misnomer for a data object.

data object type
Element

abbreviated name
RELTD_NAME

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
None

notes
A related name is a term that is in common usage; the nonstandard names of fields in existing applications, purchased software, or the names used in external standards.

Related names should not be used in place of the business name or abbreviated name of a business data object. Related names provide a means for cross-referencing between the alias and the standardized name, but this does not sanction the continued use of the alias in place of the standardized name.

derivation narrative
None

default action narrative
None

element edit string
X(31)

data format type
Text

maximum character count
31

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

RELATED NAME CONTEXT

business name
Related Name Context

description
The context or environment where a related name is used.

data object type
Element

abbreviated name
RELTD_NAME_CNTXT

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
Although this standard definition establishes no controls on the related name context, care should be taken to use the existing name of a context rather than identifying it under a different name incorrectly. Procedures and tools to aid in achieving this may implement additional rules not found in this standard.

notes
The related name context describes where a related name is meaningful. For example, if the related name is the nonstandard name of a field in an existing application, then the name of the application becomes the context.

The related name context serves to group together standardized data objects according to where they appear in some unstandardized context. Aside from application names, other examples of context include external standards, existing internal forms, third-party packages, and Digital products.

derivation narrative
None

default action narrative
None

element edit string
X(40)

data format type
Text

maximum character count
40

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

RELATIVE DATE INDICATOR

business name
Relative Date Indicator

description
A flag that indicates whether the date/time stored in a *Date* element is an absolute date/time or a relative date/time.

data object type
Element

abbreviated name
RELTV_DT_INDCR

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The relative date indicator must be one of the values listed in the following table.

Table 5–4 Relative Date Indicator Table

code values	code meaning
Relative	The date represents a period of time
Absolute	The date is an actual date/time

notes
Relative date or time is an expression of elapsed time, or of a fixed time length. For example in the expression, "one hour from now," the relative time is one hour. Relative date/time is often used for recording the actual elapsed time of some process (for example, actual time taken to inspect a shipment of parts), or for establishing schedule periods (for example frequency of sales summary reporting). In certain implementation contexts, relative date/time information is computed by subtracting the ending date/time from the starting date/time.

An absolute date/time is a specific calendar date and/or clock time.

derivation narrative
None

default action narrative
The default is *Absolute*.

element edit string
X(8)

data format type
Text

maximum character count
8

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

REPLACEMENT NARRATIVE

business name
Replacement Narrative

description
The explanation of the reasons or background for the replacement of one data object by another.

data object type
Element

abbreviated name
RPLCMT_NRTV

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The replacement narrative is basically unrestricted text that deals with the replacement of one data object by another. If other topics need to be addressed beyond the facts of replacement, then other avenues should be sought to record these topics.

notes
The replacement narrative records the business information regarding the replacement of one data object by another. Frequently, business changes will dictate that one data object be replaced by two distinct objects. The replacement narrative indicates the conditions for determining which migration path an individual occurrence of the original should follow.

The replacement narrative should address the concerns raised when a single object (for example, an element) that is used on many other objects (such as various optimized entities) is replaced. The narrative should indicate which of the applications must migrate immediately, and which at a less urgent pace.

derivation narrative
None

default action narrative
None

element edit string
T(60)

data format type
Text

maximum character count
64K

variable size indicator
Variable

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

REVERSE RELATIONSHIP NAME

business name
Reverse Relationship Name

description
The business name of a relationship as it should be read when the direction of the relationship is reversed.

data object type
Element

abbreviated name
RVRS_RLNSP_NAME

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The reverse relationship name must conform to the same rules that a normal business name of a relationship must conform to. These rules are defined in the DDNG.

The choice of the primary direction (that is, the one which is the basis for the relationship's business name) is usually based on how the relationship is commonly seen from a business viewpoint. Conversely, the reverse direction is simply the direction that is not primary. If the relationship involves more than two entities, the reverse relationship name should be drawn from the second-most commonly seen viewpoint.

The reverse relationship name may be left blank when creating standard data definitions of relationships.

notes
The reverse relationship name is primarily a means to clarify or enhance the identity of a relationship. As such, it is most valuable when it is based on a different verb than the primary business name of the relationship. For example, a relationship that has the business name *Is Based On* might easily have the reverse relationship name *Is the Basis For*. However, it would be more useful to choose a different verb phrase that is particularly appropriate to the modeling situation, such as *Has Local Variant*.

derivation narrative
The reverse relationship name is created using the procedures defined in the DDNG.

default action narrative
The initial value is blank.

subclass of

Business Name

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

SECURITY CLASSIFICATION NAME

business name
Security Classification Name

description
The name of a Proprietary Information Security Classification. This name is also used as the label on all reports, displays, and materials containing proprietary information.

data object type
Element

abbreviated name
SCURTY_CLSFTN_NAME

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The security classification name must be one of the values from the following table.

Table 5–5 Security Classification Name Table

Valid Classifications
Digital Personal
Digital Restricted Distribution
Digital Confidential
Digital Internal Use Only
Non-Proprietary

notes
Information Security Classifications are defined in section 8.03, Proprietary Information Protection, *U.S. Personnel Policy and Procedures Manual*. These policies define the classifications, requirements, and responsibilities for handling information in each classification.

These policies also state that the name of the classification that must be printed or displayed along with any proprietary information. If the classification is *Non-Proprietary*, then no special labeling is required.

derivation narrative

The security classification of a piece or collection of data must be at least as restrictive as the most restrictive classification of any of the data it contains. Also, the classification of the collection can be higher than any of its components because the combination itself makes the information more sensitive.

For example, the element *Job Code* might itself be *Digital Internal Use Only*. Similarly, the element *Employee Identifier* might be *Digital Internal Use Only*. However, when these two elements are combined in some entity, the classification of that entity might be *Digital Personal*, which would be higher than either of the elements it contains.

default action narrative

The most common security classification is *Digital Internal Use Only*, and so this can be used as the initial value. Note that it is a common error to assume that *Non-Proprietary* is the default security classification. Refer to the policies cited above for further information.

element edit string

X(32)

data format type

Text

maximum character count

32

variable size indicator

Fixed

text justification

Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

SIGNED NUMERIC INDICATOR

business name
Signed Numeric Indicator

description
The indication of whether a particular numeric element can have negative numbers as its values.

data object type
Element

abbreviated name
SIGND_NUMRC_INDCR

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The signed numeric indicator must contain a valid value from the following table.

Table 5–6 Signed Numeric Indicator Table

indicator values
Signed
Unsigned

notes
An unsigned numeric element can accept values greater than or equal to 0. Negative numbers can only be stored in a signed numeric element.

derivation narrative
None

default action narrative
The default initial value is *Unsigned*.

element edit string
X(8)

data format type
Text

maximum character count
8

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

TEXT JUSTIFICATION

business name
Text Justification

description
The rule for how the value of a textual element is to be aligned, when the size of the value is less than the size of the element.

data object type
Element

abbreviated name
TEXT_JSTFTN

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The text justification must be one of the values listed in the following table.

Table 5–7 Text Justification Table

valid values
Left
Right
Center

notes
The text justification states how to align values of an element when the size of the value is less than the size of the element. When the alignment is complete, the remaining character positions in the element will be filled with spaces.

The text justification also defines how to truncate a value when the length of that value is longer than the length of the element. The alignment will take place as defined and the portion of the text that extends beyond the size of the element will be truncated.

The text justification becomes the default field justification for the standard fields of textual elements.

derivation narrative
None

default action narrative
The default initial value is *Left*.

element edit string
X(6)

data format type
Text

maximum character count
6

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

TRANSFORMATION NARRATIVE

business name
Transformation Narrative

description
A narrative describing the rules and/or exceptions that are applied during the transformation of one data object into another.

data object type
Element

abbreviated name
TRNSFMN_NRTV

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The transformation narrative must be supplied whenever a transformation is established (that is, for every occurrence of the relationship transforms into). This narrative should explain the context, perspective, and goals of the transformation.

notes
The transformation narrative provides a place to record the business information about the transformation of one data object into another.

A transformation allows an existing data object to be redefined in a new modeling circumstance. For example, a 3NF model is transformed into an optimized model. Transformations always involve the application of human judgment, in a nonmechanistic adaptation of the information from one context to another. The criteria, perspectives, and goals of this adaptation always color the results. Therefore, it is mandatory that the transformation narrative always be supplied for every transformation.

When the original parent data object definition is changed, any data objects that are transformed from it must be evaluated for potential impact. If the child object definition does not change, a new transformation narrative should be written that links the new parent with the unchanged child.

derivation narrative
None

default action narrative
The initial value and missing value are "None."

element edit string
T(60)

data format type
Text

maximum character count
64K

variable size indicator
Variable

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

VALUE TYPE INDICATOR

business name
Value Type Indicator

description
A flag indicating whether a particular business term is a *Value Type*.

data object type
Element

abbreviated name
VAL_TYPE_INDCR

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The DDNG defines the role played by Value Types within element names. Refer to that document for complete information.

The value type indicator must be a valid value from the following table.

Table 5–8 Value Type Indicator Table

code values	code meaning
Y	The business term is a Value Type
N	The business term is not a Value Type

notes
A *Value Type* indicates what kind of data is held in an element (for example, *Quantity* and *Name* are value-type terms). As defined in the DDNG, the *business names* of all elements should end in a business term that is a Value Type. See the DDNG for further information. The *value type indicator* flags those business terms that are recommended value types.

derivation narrative
None

default action narrative
The default initial value is N.

element edit string
X

data format type
Text

maximum character count
1

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

VARIABLE SIZE INDICATOR

business name
Variable Size Indicator

description
An indicator that specifies whether a textual data item size is Variable or Fixed.

data object type
Element

abbreviated name
VARIBL_SIZE_INDCR

controller name
Enterprise Data Board

data object revision
Revision 1

data object status
Approved

integrity narrative
The variable size indicator must be one of the values listed in the following table.

Table 5–9 Variable Size Indicator Table

code values	code meaning
Fixed	The element has a fixed length.
Variable	The element has a variable length.

notes
The size of a textual element can be fixed or variable. A fixed-size element is one whose length does not change, regardless of the number of significant characters held in it. A variable-size element is one whose length varies depending on the number of characters contained in it.

If an element size is variable, the maximum character count indicates the maximum length of data values for the element.

The editing symbol "T" (known in CDO as the *long text character*) may be used in the element edit string of variable-length textual data. When it is used, the repetition count in the element edit string defines the maximum number of characters that will be printed on one line. If the total number of characters in an individual value exceeds the length specified in the repetition count, the line will be broken at the preceding word boundary and display will continue on the next line. Note that trailing spaces are not retained or displayed when the editing symbol "T" is used.

If an element has a variable size, the fields that implement it may be either variable or fixed, depending on the capabilities of the host DDL and the uses to which the field will be put. The standard fields that implement a variable size element must be able to accept the maximum size of the data item.

derivation narrative
None

default action narrative
The default initial value is *Fixed*.

element edit string
X(8)

data format type
Text

maximum character count
8

variable size indicator
Fixed

text justification
Left

security constraints

subject	security classification
Security of Data Values:	Digital Internal Use Only
Security of Metadata:	Digital Internal Use Only

A REFERENCED DOCUMENTS

A.1 EL-Class Digital Documents

EL-Class Number	Document Title
EL-CP065-00	*Data Description and Naming Guidelines*
EL-SM065-00	*Digital Standard Data Definitions Manual, proposed*

Use VTX SMC for status and to order copies of EL-class documents from Standards and Methods Control. Send distribution questions to JOKUR::SMC or call DTN: 234-4423.

A.2 Other Digital Documents

Part Number	Document Title
AA-KL45A-TE	*VAX CDD/Plus Common Dictionary Operator Reference Manual*
	DIM&T Policies and Procedures Manual
EL-EN541-00	*DMR Life Cyle System Development Guides*

To order Digital software documentation in the United States, send an approved Internal Software Order Form to USASSB Order Processing, WMO/E15. International orders must be placed with the local Digital subsidiary or approved distributor. All current DIM&T Policies and Procedures can be viewed using the VTX IM&T Infobase.

EL-EN541-00 has been inactivated. The information is now available from Educational Services, CECV03::MAILPO, Part Number EY-4424E-SG.

Index

A

Abbreviated Name • 1–7 ff, 1–15, 2–9, 2–10, 2–15, 3–2, 3–4, 3–6, 3–8, 3–12, 3–18, 3–27, 3–32, 3–34, 3–44, 3–46, 3–52, 3–55, 3–59, **5–2**, 5–62
Absolute Date • 5–92
Access Model • 3–46
Aggregate Data • 3–33, 4–2
Aggregate of • 2–16, **4–2**, 5–4
Aggregation Narrative • 1–9 ff, 2–16, 4–2, **5–4**
Alias • 3–49, 5–88
Antecedent Value • 5–40
Approval of Data Models • 1–5, 1–6, 5–36
Approved status • 5–34, 5–36
Associates • 2–11, **4–4**
Association • 1–11, 2–9, 2–15, **3–6**, 4–12, 4–33
  Subtypes of • 4–31
Atomic Data • 3–33, 4–2
Attribute • **1–17**
Attributes • 1–10, 3–3, 3–19, 3–44, 3–55, 4–21, 4–24, 4–53

B

Based on • 1–15, 2–11, 3–40, **4–6**, 4–25, 4–58
Business Data Object • 1–7 ff, 2–15, **3–8**, 4–8, 4–15, 4–27, 4–29, 4–39, 4–49, 4–51, 4–62, 5–44
  Subtypes of • 4–33
Business Name • 1–7 ff, 1–15, 2–9, 2–10, 2–15, 3–2, 3–4, 3–6, 3–8, 3–10, 3–12, 3–18, 3–27, 3–32, 3–34, 3–44, 3–46, 3–52, 3–55, 3–59, 4–51, 5–2, **5–6**, 5–14, 5–44, 5–60, 5–64, 5–96, 5–106
Business Term • 1–7 ff, 2–15, 2–16, **3–10**, 3–10, 4–8, 4–51, **5–8**, 5–8, 5–44, 5–106
Business Term Abbreviation • 2–16, 3–10, **5–10**
Business Term Definition • 2–16, 3–10, **5–12**, 5–12
Business Term Glossary • **1–18**, 3–10, 5–2, 5–6, 5–8, 5–10, 5–12, 5–44, 5–106

C

Cardinality • 4–5
  See also Maximum Cardinality
  See also Minimum Cardinality
  Symbol for • 2–2, 5–78, 5–82
CDD/Plus
  See CDO
CDO • 3–23, 3–58, 4–6, 4–16, 4–18, 5–20, 5–28, 5–48, 5–52, 5–54, 5–68
Child Element • 4–58
Classification • 1–10, 1–11, 2–15, 3–6, **3–12**, 4–31, 4–58, 5–76
Classified by • 2–16, **4–8**, 4–51
Code Description • **5–16**
Code Meaning • 2–16, 3–16
Code Set • 1–16, 2–15, 2–16, **3–14**, 4–11, 4–58, 4–60, 4–64
Code Set Name • 2–16, 3–14, **5–14**, 5–72, 5–74
Code Value • 2–15, 2–16, **3–16**, 3–16, 4–11, 5–16, **5–18**
Common Dictionary Operator
  See CDO
Complex Relationship • 1–11, 2–15, **3–18**, 3–55, 4–21, 4–23, 4–31, 4–47, 4–62, 5–38
Compound Document Data • 5–30
Computed-by Expression • 1–16, 2–15, 3–36, 3–40, 3–57, 4–6, **5–20**, 5–42
Computed-by Routine • 1–16, 3–36, 3–40, 3–57, 4–6, 5–42
Computed-by Routine Name • 2–15, **5–22**
Connector line • 2–1, 4–5
Consists of
  Code Set • 2–16, **4–10**
  Data Model • 2–11, **4–12**
Controlled by • 2–16, **4–14**
Controller • 1–8 ff, 2–15, 2–16, 3–8, 3–14, 3–16, **3–21**, 4–2, 4–10, 4–15, 4–18, 4–58, 4–64, 5–16, 5–18, 5–24, 5–26
  Responsibilities • 1–5
Controller Contact Employee Identifier • 2–16, 3–21, **5–26**
Controller Name • 2–16, 3–21, **5–24**
Created for • 2–16, **4–16**

D

Data Base Administrator • 1–6
Data Description and Naming Guidelines
  See DDNG
Data Description Language • 1–15, 2–15, 2–16, **3–23**, 4–16, 4–18
Data Description Language Name • 2–16, 3–23, **5–28**
Data Flow Diagram • 1–4
Data Form • 4–10, 4–25, 4–58
Data Format • 1–14, 2–15, **3–25**, 3–31, 3–42, 3–61, 5–30
  Subtypes of • 4–35
Data Format Type • 2–15, 3–25, 3–31, 3–42, 3–61, 4–35, 5–18, **5–30**, 5–48, 5–52
Data Item • 3–33, 3–37, 5–40
Data Model • 1–7, 2–15, **3–27**, 4–12, 4–33, 5–76
  Approval of • 1–5, 1–6
  Conceptual • 1–4
  Functional • 1–4
  Physical • 1–4
  Subtypes of • 4–37
Data Object • 2–15, **3–29**, 4–2, 4–55
  Subtypes of • 4–39
Data Object Identifier • 1–9 ff, 2–9, 2–10, 2–15, 3–2, 3–4, 3–6, 3–8, 3–12, 3–18, 3–27, 3–29, 3–32, 3–34, 3–36, 3–38, 3–40, 3–44, 3–46, 3–52, 3–55, 3–57, 3–59, **5–32**, 5–34
Data Object Revision • 1–9 ff, 2–9, 2–10, 2–15, 3–2, 3–4, 3–6, 3–8, 3–12, 3–18, 3–27, 3–29, 3–32, 3–34, 3–36, 3–38, 3–40, 3–44, 3–46, 3–52, 3–55, 3–57, 3–59, 5–32, **5–34**
Data Object Status • 1–9 ff, 2–9, 2–10, 2–15, 3–2, 3–4, 3–6, 3–8, 3–12, 3–18, 3–27, 3–29, 3–32, 3–34, 3–36, 3–38, 3–40, 3–44, 3–46, 3–52, 3–55, 3–57, 3–59, 4–55, **5–36**
Data Object Type • 2–9, 2–10, 2–15, 3–2, 3–4, 3–6, 3–8, 3–12, 3–18, 3–27, 3–29, 3–32, 3–34, 3–36, 3–38, 3–40, 3–44, 3–46, 3–52, 3–55, 3–57, 3–59, 4–31, 4–33, 4–37, 4–39, 4–41, 4–43, 4–45, 4–47, 5–6, **5–38**
Data Resource Specialists
  Responsibilities • 1–6
Data Source • 5–76
Datatype • 3–31, 3–42, 5–52
  See also Field Datatype
Date Data • 5–48, 5–92
Date Format • 2–15, 3–25, **3–31**, 4–35, 5–30
DDL
  See Data Description Language
DDL Name
  See Data Description Language Name
DDNG • 1–3, **1–18**, 3–8, 3–10, 4–8, 4–51, 4–58, 5–2, 5–6, 5–8, 5–10, 5–12, 5–44, 5–62, 5–106
Decimal Point • 5–50, 5–66
Default
  Antecedent Value • 5–40
  Edit String • 5–48
  Initial Value • 5–40, 5–70
  Missing Value • 5–40, 5–84
Default Action Narrative • 1–13, 1–16, 2–15, 3–32, **5–40**, 5–70, 5–84
Defined as • 1–15, 2–11, 4–16, **4–18**
Denormalization • 3–46, 4–58
Dependence
  Full • 3–55
  Partial • 3–2
  Transitive • 3–2
Derivation Narrative • 1–13, 1–16, 2–15, 3–32, 5–20, **5–42**
Derived Data • 3–33, 3–44, 5–42
Description • 1–7 ff, 2–9, 2–10, 2–15, 3–2, 3–4, 3–6, 3–8, 3–12, 3–18, 3–27, 3–32, 3–34, 3–44, 3–46, 3–52, 3–55, 3–59, 4–51, 5–6, **5–44**, 5–86
DIM&T Policy • 1–3
Display Format • 5–48
Documentation Source Citation • 1–15, 2–16, 3–23, **5–46**

E

EDB
  Responsibilities • 1–3, 1–5
Element • 1–13, 2–9, 2–15, **3–32**, 3–41, 4–19, 4–21, 4–24, 4–25, 4–33, 4–53, 4–58, 4–64, 5–38, 5–76
Element Edit String • 1–14, 1–16, 2–15, 3–25, 3–31, 3–36, 3–42, 3–58, 3–61, **5–48**, 5–54, 5–76, 5–108
Element Scale • 1–14, 1–16, 2–15, 3–42, **5–50**, 5–66
Ellipse • 2–1
Employee • 5–24, 5–26
Employee Identifier • 5–26
Enterprise Data Board
  See EDB
Entity • 1–9, 2–10, 2–15, **3–34**, 4–12, 4–24, 4–33, 4–53

Entity (Cont.)
Subtypes of • 4–41
Symbol for • 2–1
Entity-Relationship Diagram • 1–8
Exclusive Relationships • 1–11, 2–2, 5–76
Expressions • 3–42, 5–20
External Entity • 3–6
External Standards • 3–49, 3–51, 5–88, 5–90

F

Facets • 4–2
Field • 1–15, 2–15, **3–36**, 3–38, 4–16, 4–45
See also Nonstandard Field
See also Standard Field
Subtypes of • 4–43
Field Datatype • 1–15, 2–15, 3–36, 3–40, 3–57, 4–6, 5–30, **5–52**
Field Edit String • 1–16, 2–15, 3–36, 3–40, 3–57, 4–6, 5–48, **5–54**
Field Help • 2–15, 3–36, 3–40, 3–57, 4–6, **5–56**
Field Justification • 1–16, 2–15, 3–36, 3–40, 3–57, 4–6, **5–58**, 5–102
Field Label • 1–15, 2–15, 3–36, 3–40, 3–57, **5–60**
Field Name • 1–15, 2–15, 3–36, 3–40, 3–57, 4–6, 5–2, **5–62**
Field Query Name • 1–16, 2–15, 3–36, 3–40, 3–57, 4–6, 5–6, **5–64**
Field Scale • 1–16, 2–15, 3–36, 3–40, 3–57, 4–6, 5–50, **5–66**
Field Size • 1–16, 2–15, 3–36, 3–40, 3–57, 4–6, **5–68**, 5–80
File • 3–38
Foreign Key • 3–44, 3–55, 4–58, 4–62
Format of Data • 5–48
Full Dependence • 3–55
Functional Model • 3–46

G

Glossary
See Business Term Glossary
Graphic Data • 3–25
Group Data • 3–33, 3–44, 4–2
Group Elements • 1–14, 4–2
Group Fields • 4–2

H

Has Attribute
Complex Relationship • 2–11, **4–21**
Entity • 2–11, **4–23**
Has Data Form • 2–16, **4–25**, 4–58
Has Data Security • 2–16, **4–27**
Has Format • 5–30
Has Metadata Security • 2–16, **4–29**
Homonyms • 3–10, 5–10, 5–12

I

Image Data • 5–30
Implementation Data Object • 2–15, 3–37, **3–38**, 4–15, 4–39
Subtypes of • 4–45
Implementation of Definitions • 3–13, 3–41, 3–46, 3–55, 3–58, 3–60, 4–6, 4–18, 5–40, 5–62
Index • 3–38
Information Engineering technique • 3–52
Information Systems Managers Responsibilities • 1–6
Inheritance • 1–10, **1–18**, 2–4, 2–13, 3–7, 4–32
Initial Value • 1–16, 2–15, 3–36, 3–40, 3–57, 4–6, 5–40, **5–70**, 5–84
Input Format • 5–48
Input Validation Expression • 1–16, 2–15, 3–36, 3–40, 3–57, 4–6, 5–48, **5–72**
Input Validation Routine • 1–16, 3–36, 3–40, 3–57, 4–6
Input Validation Routine Name • 2–15, **5–74**
Integrity Narrative • 1–8 ff, 2–9, 2–10, 2–15, 3–2, 3–4, 3–6, 3–8, 3–12, 3–18, 3–27, 3–32, 3–34, 3–35, 3–44, 3–46, 3–52, 3–55, 3–59, 4–58, 4–64, 5–48, 5–70, 5–72, 5–74, **5–76**, 5–86
Interchange Format • 3–58, 5–48
Is a Business Data Object Subtype • **4–33**
Is a Data Format Subtype • **4–35**
Is a Data Model Subtype • **4–37**
Is a Data Object Subtype • **4–39**
Is a Field Subtype • **4–43**
Is an Association Subtype • **4–31**
Is an Entity Subtype • **4–41**
Is an Implementation Data Object Subtype • **4–45**
IS-A Relationship • 3–60
Is a Relationship Subtype • **4–47**

K

Key
See Foreign Key
See Primary Key
Known as • 2–16, **4–49**

L

Label
see Field Label
Security Classification • 5–98
Link • 3–55

M

Mandatory Participation • 2–2, 5–82
Many-to-Many Relationship • 2–3, 3–18, 5–78
Maximum Cardinality • 1–12, 2–3, 2–11, 3–55, 4–5, **5–78**
Maximum Character Count • 1–14, 1–16, 3–25, 3–42, 3–61, 5–50, 5–68, **5–80**
Maximum Size • 5–108
Merise technique • 1–4, 2–1, 3–46, 3–52, 4–5
Minimum Cardinality • 1–12, 2–2, 2–11, 4–5, **5–82**
Missing Value • 1–16, 2–15, 3–36, 3–40, 3–57, 4–6, 5–40, **5–84**
Modeling Symbols • 2–1, 2–2, 2–3, 3–52

N

Named Using • 2–16, **4–51**
Negative Numbers • 5–100
Nonstandard Field • 1–15, 2–10, 2–15, 3–25, 3–36, **3–40**, 4–6, 4–16, 4–43, 4–58, 5–38, 5–54, 5–62, 5–68
Notes • 1–8 ff, 2–9, 2–10, 2–15, 3–2, 3–4, 3–6, 3–8, 3–12, 3–18, 3–27, 3–32, 3–34, 3–44, 3–46, 3–52, 3–55, 3–59, **5–86**
Numeric Data • 5–48, 5–100
Numeric Format • 2–15, 3–25, **3–42**, 4–35, 5–30, 5–50

O

Obsolete status • 4–55, 4–62, 5–34, 5–36
One-to-Many Relationship • 5–78
One-to-One Relationship • 5–78
Optimization • 3–44, 3–46, 5–104
Optimized Entity • 1–9, 2–15, **3–44**, 4–23, 4–41, 4–62, 5–38
Optimized Model • 1–4, 1–7, 2–10, 2–15, 3–19, **3–46**, 3–52, 4–37, 4–62, 5–38
Optimized Model Requirements • 1–7
Optional Participation • 2–2, 5–82

P

Parent Element • 4–58
Partial Dependence • 3–2
Participates in Primary Key of • 2–11, **4–53**
Primary Key • 1–10, 3–2, 3–19, 3–35, 4–23, **4–53**
Property • 4–21, 4–24
Punctuation • 5–2

R

Record • 3–38
Recursive Relationship • 3–6, 3–52, 3–55
Related Name • 2–15, 2–16, **3–49**, 3–49, 4–49, 5–14, **5–88**
Related Name Context • 2–15, 2–16, **3–51**, 4–49, **5–90**
Related Names • 1–8 ff
Relationship • 1–11, 2–15, **3–52**, 4–31, 5–76, 5–96
Complex • 3–18, 3–55
Exclusive • 5–76
Many-to-Many • 3–52, 3–55, 5–78
One-to-Many • 5–78
One-to-One • 5–78
"Or" Participation • 2–2
Recursive • 3–52, 3–55
Simple • 3–55
Subtypes of • 4–47
Symbol for • 2–1, 3–52
Relative Date • 5–92
Relative Date Indicator • 1–14, 2–15, 3–31, **5–92**

Repeating Groups • 3–2, 3–44
Replacement Narrative • 1–9 ff, 2–16, 4–55, **5–94**
Replaces • 2–16, **4–55**, 5–34, 5–94
Reverse Relationship Name • **5–96**
Revision • 4–49, 4–55, 4–62, 5–34, 5–36, 5–94
Root Element • 4–58
Roundtangle • 2–1

S

Scope of a Data Model • 5–76
Security • 3–44, 3–46, 5–98
Security Classification • 1–8 ff, 2–15, 2–16, 3–8, **3–54**, 4–27, 4–29
Security Classification Name • 2–16, 3–54, **5–98**
Segment • 4–2, 5–4
Semantic Relationship
  see Subclass of
  See Subtyping
Signed Numeric Indicator • 1–14, 2–15, 3–42, **5–100**
Simple Relationship • 1–11, 2–10, 2–15, **3–55**, 4–31, 4–47, 5–38
Size in Characters • 1–14, 2–15, 5–108
Size in Digits • 2–15
SLC
  See System Development Life Cycle
Source of Data • 5–76
Standard Data Definitions • 1–1, **1–2**
Standard Data Model Requirements • **2–6**, 3–4
Standard Field • 1–15, 2–15, 3–41, **3–57**, 4–6, 4–16, 4–19, 4–43, 5–2, 5–38, 5–62
Standardization Process • 1–3, 1–5, 2–6, 2–12
Standards
  External • 3–49, 3–51, 5–88, 5–90
Storage Format • 5–48
  See also Datatype
Subclass of • 1–14, 2–16, 3–14, 3–26, 4–10, 4–25, **4–57**, 4–60, 4–64, 5–16
Subfields • 4–2
Subject Area Model • 1–4
Subject Areas • 4–2, 4–8
Subroutine • 5–22
Subset of • 2–16, 4–2, 4–10, **4–60**, 4–64
Subtype • 2–4, 3–60
Subtyping • 1–10, 1–11, 2–13, 2–15, 3–6, **3–59**, 4–31, 4–58, 4–62, 5–38, 5–76
  Symbol for • 2–3
Supertype • 2–4, 3–60
Synonyms • 3–10, 5–10, 5–12, 5–88
System Development Life Cycle • 1–3, 1–6, 2–6, 2–12

T

Tables • 3–14
Text Data • 5–48, 5–102, 5–108
Text Format • 2–15, 3–25, **3–61**, 4–35, 5–30
Text Justification • 1–14, 1–16, 2–15, 3–61, 5–58, **5–102**
Third Normal Form • 1–2, **1–18**, 3–3, 3–32, 4–23
  3NF Entity • 1–9, 2–9, 2–15, **3–2**, 4–23, 4–41, 4–53, 5–38
  3NF Model • 1–4, 1–7, 2–9, 2–15, 3–4, 3–52, 3–56, 4–37, 5–38
  3NF Model Requirements • 1–7, 3–4
Third Party
  Applications • 3–49, 3–51, 5–88, 5–90
  Data Base Managers • 3–23
  Query Tools • 3–23
Time Data • 3–31, 5–92
Transformation • 2–7, 3–4, 3–13, 3–19, 3–44, 3–46, 3–60, 4–58, 4–62, 5–104
Transformation Narrative • 1–8 ff, 2–11, 2–16, **5–104**
Transforms into • 2–7, 2–11, 2–16, 3–19, **4–62**, 5–104
Transitive Dependence • 3–2
Type Hierarchy of Data Objects • 2–3, **2–12**, 3–29, 5–38

U

Unsigned Numeric • 5–100

V

Values Bounded by • 2–16, **4–64**
Value Type • 3–10, 5–6, 5–106
Value-Type Indicator • 2–16, 3–10, **5–106**
Variable Length Data • 3–61
Variable Size Indicator • 1–14, 2–15, 3–61, 5–80, **5–108**
VAX CDD/Plus

VAX CDD/Plus (Cont.)
See CDO
Voice Data • 3–25, 5–30

W

Weak Entity • 1–9, 3–35, 3–60, 4–53, 5–76
Work in Process status • 5–34, 5–36

Y

Yourdon technique • 3–52

EL-00065-00 **08-Mar-1991**

```
+-------------------------------------------------------------------+
|                          READER COMMENTS                          |
|                                                                   |
| Your  comments  and suggestions  will help  Standards and Methods |
| Control improve their services and documents.                     |
+-------------------------------------------------------------------+
```

Did you request this document? _____ If so, did it arrive within a satisfactory period of time? _____ Please comment.

What are your impressions of this document? Consider format, organization, completeness, readability, and illustrations.

————————**FOLD ON THIS LINE**————————————————————

Did you find technical or clerical errors in this document? If so, please specify the page number(s) and the error(s).

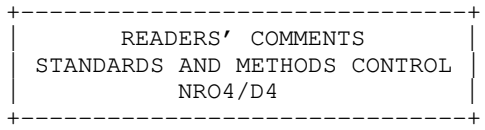
Are the instructions for the update package clear? _____
Was an index available? _____ If not, is one needed? _____
Do you have other suggestions for improving this document?

The following information is optional:

Name _____ Mailstop _____

Department _____ Node _____

Send your comments to JOKUR::PROJECTS, or fold, staple, and send this page through interoffice mail to:


READERS' COMMENTS
STANDARDS AND METHODS CONTROL
NRO4/D4